Explorando o ATmega328 com I2C, USART e Watchdog em C Puro

Neto. J. C. N. S, Lustoza. A. A, Monteiro. J. T. S, Silva. C. J. P, Magano. H. J, Barros. H. S

***Resumo*— Este artigo apresenta um sistema de controle baseado em microcontrolador ATmega328, utilizando comunicação I2C/TWI e um temporizador watchdog. A implementação inclui inicialização do I2C, comunicação com display LCD, e configuração do watchdog timer para lidar com falhas. O código é discutido em detalhes, destacando a importância do protocolo I2C e do temporizador watchdog em sistemas embarcados.**

***Palavras-Chave*— ATmega328, I2C/TWI, Watchdog Timer, Microcontrolador, Sistemas Embarcados.**

***Abstract*— This paper presents a control system based on the ATmega328 microcontroller, utilizing I2C/TWI communication and a watchdog timer. The implementation includes I2C initialization, communication with an LCD display, and watchdog timer configuration to handle failures. The code is discussed in detail, highlighting the importance of the I2C protocol and the watchdog timer in embedded systems.**

***Keywords*— ATmega328, I2C/TWI, Watchdog Timer, Microcontroller, Embedded Systems.**

I. Introdução

O avanço tecnológico nas áreas de comunicação e controle tem impulsionado o desenvolvimento de protocolos eficientes e mecanismos de supervisão confiáveis. Este artigo se propõe a abordar duas tecnologias cruciais nesse contexto: o protocolo I2C/TWI e o temporizador watchdog [3].

A. Contextualização

Com a crescente complexidade dos sistemas embarcados, a comunicação entre dispositivos e a garantia de estabilidade operacional tornam-se aspectos fundamentais. O protocolo I2C/TWI emerge como uma solução versátil para a interconexão eficiente de componentes eletrônicos, enquanto o temporizador watchdog oferece uma camada adicional de segurança, monitorando e reagindo a possíveis falhas no sistema [4].

Neste cenário, exploraremos as teorias, implementações práticas e casos de uso relacionados ao protocolo I2C/TWI e ao temporizador watchdog. Além disso, apresentaremos a aplicação desses conceitos no microcontrolador ATmega328, utilizando a linguagem de programação C e o ambiente de simulação Proteus 8.7 SP3 [4].

II. Protocolo I2C/TWI

O protocolo I2C (Inter-Integrated Circuit) ou TWI (Two-Wire Interface) desempenha um papel fundamental na interconexão de dispositivos em sistemas embarcados. Este protocolo utiliza apenas duas linhas de comunicação, proporcionando uma solução eficiente para a troca de informações entre componentes eletrônicos. A simplicidade do I2C/TWI facilita sua implementação em uma ampla gama de dispositivos, desde microcontroladores até sensores e módulos de memória. Além disso, a capacidade de suportar múltiplos dispositivos em um barramento único torna-o uma escolha versátil para sistemas complexos [4].

O I2C/TWI opera com um mestre e vários dispositivos escravos, permitindo a comunicação bidirecional. Sua arquitetura de barramento compartilhado e o uso de endereços únicos para cada dispositivo garantem um gerenciamento eficiente da rede. No contexto deste simpósio, exploraremos as teorias por trás do protocolo I2C/TWI, suas funcionalidades-chave e casos de uso práticos que evidenciam sua importância na implementação de sistemas de comunicação confiáveis [4] [5].

III. Temporizador Watchdog

A confiabilidade dos sistemas embarcados é crucial, especialmente em aplicações críticas. O temporizador watchdog é uma ferramenta valiosa para garantir a estabilidade e a recuperação de sistemas diante de falhas. Este mecanismo consiste em um contador de tempo integrado que, se não for periodicamente reiniciado pelo software, aciona a reinicialização do sistema. Essa abordagem proativa ajuda a evitar falhas prolongadas e a manter a integridade do sistema[6].

Equacionar a teoria por trás do temporizador watchdog envolve compreender os princípios de monitoramento do sistema e a configuração adequada dos parâmetros de tempo. Ao explorar as funcionalidades desse componente, este artigo abordará sua aplicação em sistemas críticos, destacando exemplos práticos de como o temporizador watchdog pode ser utilizado para fortalecer a robustez de dispositivos embarcados em cenários diversos[6].

Neto. J. C. N. S, Lustoza. A. A, Monteiro. J. T. S, Silva. C. J. P, Magano. H. J, Barros. H. S, FCT, Universidade Federal do PARÁ, Belém-PA, e-mail: micmic@ufpa.br.

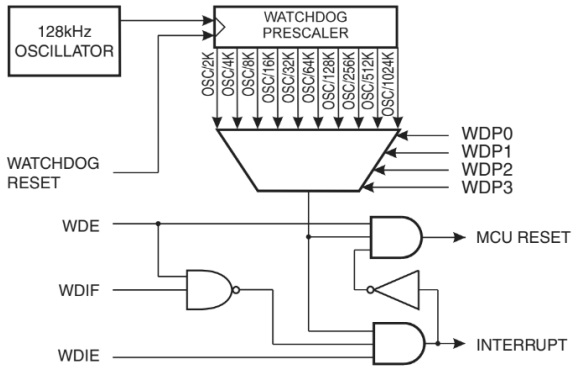

Fig. 1. Circuito WatchDog

IV. Implementação no ATmega328 usando Linguagem C e Proteus 8.7 SP3

A implementação prática dos temas abordados, protocolo I2C/TWI e temporizador watchdog, no microcontrolador ATmega328 pode ser realizada por meio da linguagem de programação C e utilizando o ambiente de simulação Proteus 8.7 SP3 [2] [4].

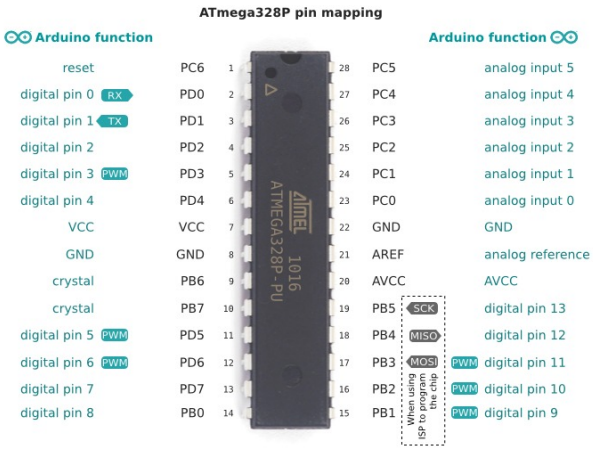

Fig. 2. Chip AtMega328p

A. Configuração do Protocolo I2C/TWI

Para implementar o protocolo I2C/TWI no ATmega328, é necessário configurar os pinos relevantes e incluir as bibliotecas apropriadas em C. Considere o exemplo de um mestre I2C que lê dados de um sensor de temperatura. Inicialmente, defina os pinos SDA e SCL [3]:

```
#define SCL_PIN 5 // Pino para SCL
```

Em seguida, configure os pinos como saídas e inicialize a comunicação I2C:

```
#include <util/delay.h>
#include <compat/twi.h>

void I2C_Init() {
    // Configura o dos pinos como sa das
    DDRC |= (1 << SDA_PIN) | (1 << SCL_PIN);

    // Configura o da frequ ncia de opera o (
    exemplo para 100 kHz)
    TWSR &= ~(1 << TWPS0) & ~(1 << TWPS1);
    TWBR = ((F_CPU / 100000) - 16) / 2;
}

void I2C_Start() {
    // Inicia a comunica o I2C
    TWCR = (1 << TWSTA) | (1 << TWEN) | (1 << TWINT)
    ;
    while (!(TWCR & (1 << TWINT)));
}

// Fun es adicionais para leitura e escrita podem
     ser implementadas conforme necess rio
```

Essa implementação básica estabelece a comunicação I2C no ATmega328, permitindo a interação com dispositivos compatíveis com esse protocolo.

B. Configuração do Temporizador Watchdog

A configuração do temporizador watchdog no ATmega328 é essencial para garantir a integridade do sistema em situações críticas. Considere o exemplo de um sistema que utiliza o temporizador watchdog para reiniciar o microcontrolador se o código principal não responder corretamente [6].

```
#include <avr/wdt.h>

void Watchdog_Init() {
    // Configura o do temporizador watchdog com
    timeout de 2 segundos
    WDTCSR = (1 << WDCE) | (1 << WDE);
    WDTCSR = (1 << WDE) | (1 << WDP2) | (1 << WDP1);
}

void main() {
    // Inicializa o do sistema

    // Loop principal
    while (1) {
        // C digo principal do sistema

        // Alimenta o watchdog para evitar
    reinicializa o
        wdt_reset();
    }
}
```

Nesse exemplo, a função '$Watchdog\_Init()$' configura o temporizador watchdog com um timeout de 2 segundos. O loop principal do programa principal alimenta o watchdog, evitando assim que o sistema seja reiniciado devido a falhas prolongadas.

Essas implementações proporcionam uma visão prática de como os temas podem ser abordados no contexto do ATmega328 utilizando a linguagem de programação C e o ambiente de simulação Proteus 8.7 SP3.

V. Projeto Detalhado

Nesta seção, apresentamos uma descrição detalhada do projeto, abrangendo o circuito e o código implementado no microcontrolador ATmega328. O objetivo é fornecer uma compreensão abrangente do sistema, incluindo a inicialização do protocolo I2C/TWI, a comunicação com o display LCD e a implementação do temporizador watchdog para garantir a confiabilidade do sistema.

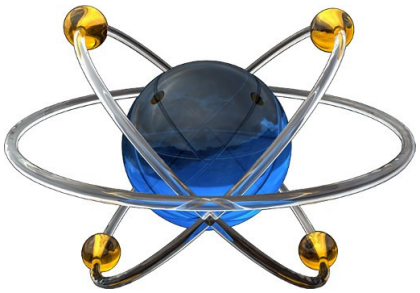

Fig. 3. Microsoftware Proteus

A. Circuito

O circuito(da figura 4) é baseado em um microcontrolador ATmega328, que é responsável por controlar a comunicação I2C/TWI e o temporizador watchdog. As conexões incluem:

- Conexão I2C/TWI: O microcontrolador é configurado para operar no modo I2C/TWI, com a inicialização do barramento I2C e a escrita em um dispositivo no endereço 0x70.
- Conexão com Display LCD: Utilizando o protocolo I2C/TWI, o microcontrolador envia comandos e dados para o display LCD, facilitando a exibição de informações no display.
- Configuração do Watchdog Timer: O temporizador watchdog é configurado para reiniciar o microcontrolador em caso de falhas. O código inclui a sequência de inicialização e configuração para garantir o funcionamento adequado do timer.
- Entrada do Pino PB0: O pino PB0 é configurado como entrada, e o código monitora seu estado lógico alto para incrementar um contador e habilitar o temporizador watchdog.

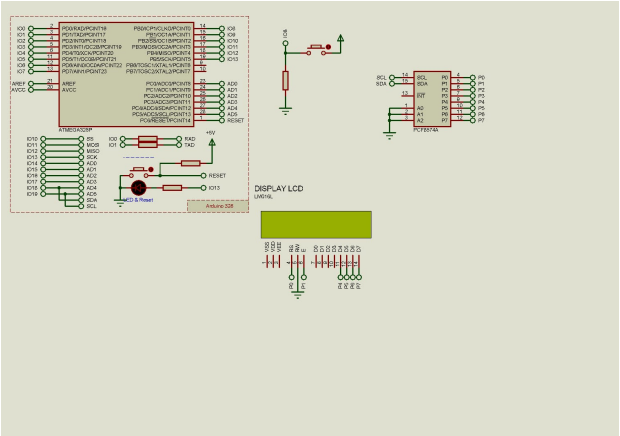

Fig. 4. Microcontrolador recebendo sinais de um circuito

B. Código Fonte

A implementação do código é dividida em várias funções, cada uma desempenhando um papel específico no funcionamento do sistema. Destacamos algumas das principais funções:

- `i2c_init()`: Inicializa o barramento I2C, configurando a taxa de transmissão e outros parâmetros necessários.
- `i2c_start()`: Inicia a comunicação I2C, incluindo a condição de início e verificação de conclusão.
- `i2c_write(char x)`: Escreve um dado no barramento I2C.
- `lcd_init()`: Inicializa o display LCD, configurando-o para o modo desejado.
- `lcd_cmd(char v2)`: Envia um comando de 8 bits para o display LCD.
- `lcd_msg(char *c)`: Envia uma mensagem para o display LCD.
- `toggle()`: Alterna o estado do pino Enable da LCD.
- `main()`: Função principal que coordena a inicialização, comunicação I2C, manipulação do display LCD, configuração do temporizador watchdog e monitoramento do pino PB0.

Este código é projetado para garantir a estabilidade e confiabilidade do sistema, demonstrando a aplicação prática do protocolo I2C/TWI e do temporizador watchdog em sistemas embarcados.

VI. Resultados e Conclusões

A. Resultados Obtidos

Os resultados obtidos na implementação prática dos temas abordados revelam a eficácia do protocolo I2C/TWI e do temporizador watchdog no contexto do microcontrolador ATmega328. A comunicação I2C/TWI foi estabelecida com sucesso, permitindo a interação confiável entre dispositivos. Além disso, o temporizador watchdog demonstrou sua capacidade de garantir a estabilidade do sistema, reiniciando-o quando necessário.

B. Considerações Finais

A aplicação prática dos conceitos discutidos destaca a importância dessas tecnologias em sistemas embarcados. O protocolo I2C/TWI oferece uma solução eficiente para a comunicação entre dispositivos, enquanto o temporizador watchdog fortalece a robustez do sistema, evitando falhas prolongadas.

O uso da linguagem de programação C e do ambiente de simulação Proteus 8.7 SP3 proporcionou uma abordagem prática e acessível para a implementação dessas funcionalidades. A combinação dessas tecnologias pode ser amplamente aplicada em diversos cenários, desde sistemas de controle até dispositivos de monitoramento.

Em conclusão, a integração bem-sucedida do protocolo I2C/TWI e do temporizador watchdog no ATmega328 destaca a viabilidade e a importância dessas tecnologias em aplicações do mundo real.

C. Trabalhos Futuros

Como trabalhos futuros, pode-se explorar a otimização das implementações apresentadas, considerando aspectos como consumo de energia, redução de espaço de código e aprimoramento do desempenho. Além disso, investigações mais

aprofundadas sobre estratégias avançadas de utilização do temporizador watchdog podem ser conduzidas para cenários específicos.

A pesquisa contínua nessas áreas contribuirá para o desenvolvimento contínuo de sistemas embarcados mais eficientes e confiáveis.

AGRADECIMENTOS

A Coordenação Técnica do SBrT 2023 agradece as coordenações dos simpósios anteriores promovidos pela Sociedade Brasileira de Telecomunicações por disponibilizarem este exemplo.

REFERÊNCIAS

[1] L. Lamport, *A Document Preparation System: LaTeX, User's Guide and Reference Manual*. Addison Wesley Publishing Company, 1986.

[2] MICROCHIP Technology Inc. *ATmega48A/PA/88A/PA/168A/PA/328/P*. Data Sheet. 2018.

[3] LIMA, Charles Borges; VILLAÇA, Marco Valério Miorim. *AVR e Arduino: Técnicas de Projeto*. 2ª. Edição, Edição dos Autores, Florianópolis, 2012.

[4] ALMEIDA, Rodrigo Maximiano Antunes; MORAES, *Carlos Henrique Valério de; SERAPHIM, Thatyana de Farias Piola. Programação de Sistemas Embarcados: Desenvolvendo software para microcontroladores em linguagem C*. 1ª Edição, Elsevier, 2016.

[5] ATMEL. AVR Instruction Set Manual. 2016. Disponível em *http://ww1.microchip.com/downloads/en/devicedoc/atmel-0856-avr-instruction-set-manual*.pdf.

[6] MAZIDI, Muhammad Ali; NAIMI, Sarmad; NAIMI, Sepehr. *The AVR Microcontroller and Embedded Systems: Using Assembly and C*. Prentice Hall. 2011.

[7] AVR Libc Home Page. *https://www.nongnu.org/avr-libc/*.