MiniProjeto - Fechadura Eletrônica

Guilherme José M. Jerônimo

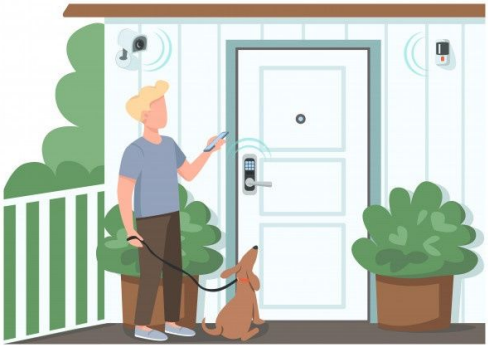
Capacitação em Embarcados 2026 - VIRTUS UFCG

Fevereiro 6, 2026

Sumário

Introdução - Motivações

- Crescente demanda por segurança e automação residencial
- Limitações de fechaduras mecânicas tradicionais
- Uso de microcontroladores como solução de baixo custo
- Aplicações reais: casas, laboratórios, armários, empresas



Características do ATmega328P

O Microcontrolador ATmega328P:

- O ATmega328P é um microcontrolador CMOS de 8 bits de alto desempenho e baixo consumo de energia, baseado na arquitetura RISC (Reduced Instruction Set Computer) avançada da AVR.

Periféricos e I/O:

- Pinos de I/O: 23 pinos programáveis.
- Canais ADC: 6 canais em encapsulamento DIP (8 canais em TQFP) com resolução de 10 bits.
- Temporizadores (Timers): Dois de 8 bits e um de 16 bits.
- Canais PWM: 6 saídas para controle de potência e sinal analógico simulado.
- Comunicação Serial: Suporte nativo para USART, SPI e I2C (TWI).

Características do ATmega328P

Vantagens para Projetos:

- PicoPower: Consumo extremamente baixo em modos de repouso (Sleep Modes).
- Popularidade: Documentação vasta e suporte total da plataforma Arduino.
- Eficiência: Relação de quase 1 MIPS por MHz, permitindo otimização entre consumo e processamento.

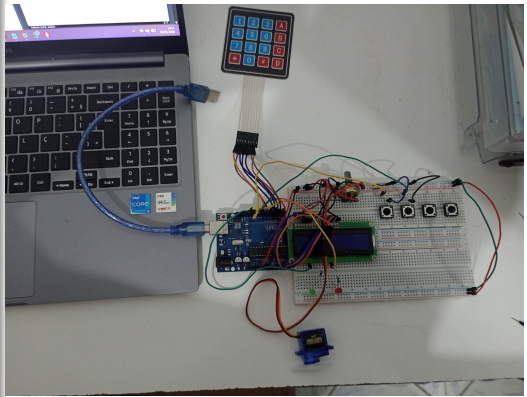
Principais Características Técnicas:

- Arquitetura: RISC de 8 bits.
- Memória Flash: 32 KB.
- SRAM: 2 KB (memória volátil para variáveis).
- EEPROM: 1 KB (armazenamento de dados não volátil).
- Frequência de Operação: Até 20 MHz.
- Tensão de Operação: 1.8V a 5.5V.

Hardware Utilizado

Descrição do sistema de fechadura eletrônica:

- ATmega328p
- LCD 1602
- Teclado Matricial
- Push Buttons
- Servo Motor
- LEDs
- Resistores
- Jumpers
- Potenciômetro
- Protoboard



Software Embarcado

GPIO

- Portas usadas

```
DDRD &= ~((1 << PD4) | (1 << PD3) | (1 << PD2));
PORTD |= (1 << PD4) | (1 << PD3) | (1 << PD2);

DDRB |= (1 << PB0) | (1 << LED_PINVERD) | (1 << LED_PINVERM);
DDRD |= (1 << PD7) | (1 << PD6) | (1 << PD5);

DDRB &= ~((1 << PB2) | (1 << PB3));
PORTB |= (1 << PB2) | (1 << PB3);
```

Timer e PWM

- Timer → cria o tempo base (20 ms)
- PWM → usa esse tempo para gerar pulsos
- Servo → lê a largura do pulso e gira

```
void init_servo(void) {
    DDRB |= (1 << SERVO_PIN);

    TCCR1A = (1 << COM1A1) | (1 << WGM11);
    TCCR1B = (1 << WGM13) | (1 << WGM12) | (1 << CS11);

    ICR1 = 39999;
    posicionar_servo(0);
}

void posicionar_servo(uint16_t angulo) {
    OCR1A = ((uint32_t)angulo * 2000) / 180 + 1000;
}
```

Software Embarcado

Interrupção PB2

- Botão Interno: abre a porta instantaneamente por dentro do ambiente.

```
ISR(PCINT0_vect) {
    uint8_t estadoAtual = PINB;
    uint8_t mudanca = estadoAtual ^ ultimoEstadoPB;

    if (mudanca & (1 << PB2)) {
        if (!(estadoAtual & (1 << PB2)) && (travapb2 == 0)) {
            if(eventoPB2 == 0){
                eventoPB2 = 1;
                travapb2 = 1;
            }
        }
    }
```

```
if (eventoPB2) {
    USART_SendString("\r\nInterrupcao no pb2!\r\n");

    PORTB &= ~(1 << LED_PINVERM);
    PORTB |= (1 << LED_PINVERD);

    posicionar_servo(180);
    _delay_ms(3000); // Porta fica aberta por 3 segundos
    posicionar_servo(0);

    PORTB &= ~(1 << LED_PINVERD);

    // --- NOVO RECURSO: AGUARDAR SOLTAR O BOTÃO ---
    // Enquanto o pino PB2 estiver em nível baixo (pressionado), o código "trava" aqui
    while (!(PINB & (1 << PB2))) {
        _delay_ms(10); // Pequena espera para debounce ao soltar
    }

    // Agora que o botão foi solto, limpamos a flag para permitir um novo clique
    eventoPB2 = 0;
    travapb2 = 0;

    _delay_ms(200); // Tempo de segurança adicional
}
```

Software Embarcado

Interrupção PB3

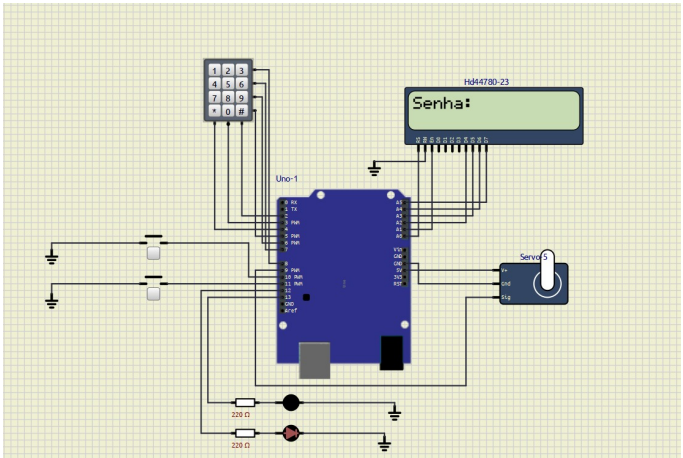
- Botão de Configuração: o usuário pode cadastrar uma nova senha de 4 dígitos.

```
if (mudanca & (1 << PB3)) {
    if (!(estadoAtual & (1 << PB3)) && (travapb3 == 0)) {
        if(eventoPB3 == 0){
            eventoPB3 = 1;
            travapb3 = 1;
        }
    }
}
```

```
if (eventoPB3) {
    USART_SendString("\r\nInterrupcao no pb3!\r\n");

    modo_cadastro = 1;
    estado = SENHA_ANTERIOR;

    buf = NULL;
    if(buf == NULL){
        USART_SendString("\r\n buf nulo\r\n");
    }else{
        USART_SendString(buf);
    }
    indice_senha = 0;
    senha_digitada[0] = '\0';
    USART_SendString("\r\n Digite Senha Anterior: \r\n");

    lcd_command(0x01);
    lcd_string("Senha antiga:");
    lcd_command(0xC0);

    eventoPB3 = 0;
    travapb3 = 0;
    _delay_ms(1000);
    continue;
```

Simulide

Fechadura Eletrônica



Possíveis Melhorias

Aprimoramentos

- RFID / Biometria
- Comunicação Bluetooth ou Wi-Fi
- Registro de acessos
- Aplicativo móvel

Conclusão

Resultados Obtidos

- Funcionamento da fechadura
- Confiabilidade do sistema
- Tempo de resposta
- Impacto do projeto
- Aprendizado técnico adquirido

Referências e Complementações

Materiais de Estudo

https://drive.google.com/drive/folders/1AsjytyM8in8946E79xPQg-f-JvHl8uiK?usp=sharing

Repositório no GitHub

https://github.com/GuiMendespy/Fechadura_Eletronica_ATmega328p