

Documentação da Biblioteca

`gpio_irq_manager`

Documentação da Biblioteca gpio_irq_manager

A biblioteca `gpio_irq_manager` foi desenvolvida para gerenciar interrupções GPIO no Raspberry Pi Pico de forma eficiente e modular. Ela permite que outras bibliotecas registrem e removam callbacks para eventos específicos em pinos GPIO, além de tratar o debounce (eliminação de ruídos) para garantir que as interrupções sejam acionadas de forma confiável.

Objetivo Principal

O objetivo principal desta biblioteca é **facilitar o gerenciamento de interrupções GPIO para outras bibliotecas**. Em vez de cada biblioteca precisar implementar sua própria lógica de interrupção, elas podem usar o `gpio_irq_manager` para registrar callbacks e lidar com eventos de interrupção de forma centralizada.

Estrutura da Biblioteca

A biblioteca é composta por dois arquivos principais:

- gpio_irq_manager.h**: Contém as declarações das funções e constantes utilizadas pela biblioteca.
- gpio_irq_manager.c**: Contém a implementação das funções declaradas no arquivo `.h`.

Funcionalidades Principais

- Registro de Callbacks**: Permite que outras bibliotecas registrem funções de callback para eventos específicos em pinos GPIO.
- Remoção de Callbacks**: Permite que outras bibliotecas removam callbacks previamente registrados.
- Tratamento de Debounce**: Implementa um mecanismo de debounce para evitar múltiplas interrupções causadas por ruídos.
- Inicialização do Gerenciador**: Configura a função de tratamento de interrupções global e habilita interrupções no banco de GPIOs.

Constantes e Variáveis Globais

Constantes

```
#define DEBOUNCE_DELAY_MS 200
```

- **Descrição:** Define o tempo de debounce em milissegundos. Este é o intervalo mínimo entre duas interrupções consecutivas para evitar ruídos.

Variáveis Globais

```
void (*callbacks[MAX_GPIO_PINS])(void) = {NULL};
```

- **Descrição:** Vetor de callbacks para eventos GPIO. Cada posição no vetor corresponde a um pino GPIO, e o valor é um ponteiro para a função de callback.

```
absolute_time_t last_interrupt_time[MAX_GPIO_PINS];
```

- **Descrição:** Array para armazenar o último tempo de interrupção para cada pino. Usado para implementar o debounce.

Funções da Biblioteca

gpio_irq_handler

Função de tratamento de interrupções GPIO. Esta função é chamada automaticamente pelo hardware quando ocorre uma interrupção GPIO.

```
void gpio_irq_handler(uint gpio, uint32_t events);
```

Parâmetros:

- **gpio**: Pino GPIO que gerou a interrupção.
- **events**: Eventos que causaram a interrupção (borda de subida, descida, etc.).

Comportamento:

- Verifica se o pino GPIO é válido e se há um callback registrado.
- Verifica se o tempo desde a última interrupção é maior que o tempo de debounce.
- Se todas as condições forem atendidas, chama o callback correspondente.

register_gpio_callback

Registra uma função de callback para um pino GPIO e um evento específico.

```
void register_gpio_callback(uint gpio, void (*callback)(void), uint32_t event_mask);
```

Parâmetros:

- **gpio**: Pino GPIO para o qual o callback será registrado.
 - **callback**: Função de callback a ser chamada quando o evento ocorrer.
 - **event_mask**: Máscara de eventos que acionarão o callback (borda de subida, descida, etc.).
-

`remove_gpio_callback`

Remove um callback registrado para um pino GPIO.

```
void remove_gpio_callback(uint gpio, uint32_t event_mask);
```

Parâmetros:

- `gpio`: Pino GPIO do qual o callback será removido.
 - `event_mask`: Máscara de eventos para a qual o callback será removido.
-

`gpio_irq_manager_init`

Inicializa o gerenciador de interrupções GPIO. Configura a função de tratamento de interrupções global e habilita interrupções no banco de GPIOs.

```
void gpio_irq_manager_init();
```

Considerações Finais

A biblioteca `gpio_irq_manager` é uma ferramenta poderosa para **centralizar o gerenciamento de interrupções GPIO** no Raspberry Pi Pico. Ela permite que outras bibliotecas registrem e removam callbacks de forma simples, sem precisar lidar diretamente com a complexidade das interrupções GPIO. Além disso, o tratamento de debounce integrado garante que as interrupções sejam acionadas de forma confiável, evitando ruídos.