

OPTA-PROG-PROCESSOR-INTERACTION

PT-BR

Interação entre núcleos do OPTA

1. Informações Gerais

- **Versão / Revisão:** *Revisão 1*
- **Data de Desenvolvimento:** 03/01/2025
- **Autor:** Daniel Arcos
- **Linguagem:** C++
- **Plataforma / Hardware:** Finder OPTA

2. Objetivo da Programação

Esta programação visa demonstrar como ocorre a interação entre os núcleos M7 e M4 do OPTA. Quando um pulso no botão de usuário do produto for interpretado no núcleo M7, esse pulso será processado e o núcleo M4 executará uma função com base nele. Neste exemplo de demonstração, o LED_D0 é ativado com base no acionamento do botão de usuário do OPTA.

3. Requisitos e Dependências

- Instalar o Arduino IDE em sua última versão disponível
- Instalar os drivers de Hardware do OPTA no Arduino IDE
- Instalar biblioteca do OPTA
- Instalar as bibliotecas descritas na programação

4. Passo a Passo da Implementação

A programação está pronta para uso, basta enviá-la para a OPTA. Lembre-se de que são utilizados 2 programas, um para o núcleo M4 e outro para o núcleo M7.

ATENÇÃO: É necessário separar as memórias e os núcleos na aba "Ferramentas" da IDE, conforme descrito abaixo:

Para a programação com o nome OPTA-PROG-PROCESSOR-INTERACTION-M7, siga estas configurações:

Tools > Flash split > 1MB M7 + 1MB M4

Tools > Target core > Main Core

Após concluir a configuração, você pode enviar a programação para o produto.

Para a programação com o nome OPTA-PROG-PROCESSOR-INTERACTION-M4, siga estas configurações:

Tools > Flash split > 1MB M7 + 1MB M4

Tools > Target core > M4 Co-processor

Após concluir a configuração, você pode enviar a programação para o produto.

5. Testes e Exemplos de Uso

Programação M7:

Funcionamento:

1. **Inicialização:** Configura o botão do usuário como entrada, inicia a comunicação RPC e inicializa o núcleo M4.
2. **Loop principal:**

- Lê o estado do botão do usuário
- Se o botão estiver pressionado (LOW), chama a função `ativarSaida()` no M4 via RPC
- Se o botão não estiver pressionado, chama `desativarSaida()` no M4
- Adiciona um pequeno delay para evitar processamento excessivo

Fluxo Completo do Sistema

1. O M7 fica constantemente monitorando o botão do usuário
2. Quando o botão é pressionado:
 - M7 chama `RPC.call("ativarSaida")`
 - M4 recebe a chamada e executa `ativarSaida()`, ligando D0 e LED_D0
3. Quando o botão é solto:
 - M7 chama `RPC.call("desativarSaida")`
 - M4 recebe a chamada e executa `desativarSaida()`, desligando D0 e LED_D0

Programação M4:

Funcionamento:

1. **Inicialização:** Configura o pino D0 e LED_D0 como saídas e inicia a comunicação RPC com o M7.
 2. **Funções registradas:**
 - `ativarSaida()` : Liga o pino D0 e o LED_D0
 - `desativarSaida()` : Desliga o pino D0 e o LED_D0
 3. **Comunicação:** As funções são registradas via `RPC.bind()` para que possam ser chamadas remotamente pelo M7.
 4. **Loop:** Vazio, pois toda ação é disparada por chamadas RPC externas.
-

Interaction between OPTA nuclei

1. General Information

- Version/Revision: Revision 1
- Development Date: 01/03/2025
- Author: Daniel Arcos
- Language: C++
- Platform/Hardware: Finder OPTA

2. Purpose of Programming

This program aims to demonstrate how the interaction between the M7 and M4 cores of the OPTA occurs. When a pulse on the product's user button is interpreted in the M7 core, this pulse will be processed and the M4 core will execute a function based on it. In this demonstration example, LED_D0 is activated based on the activation of the OPTA user button.

3. Requirements and Dependencies

- Install the latest version of Arduino IDE
- Install the OPTA hardware drivers in Arduino IDE
- Install the OPTA library
- Install the libraries described in the programming

4. Implementation Step by Step

The programming is ready to use, just send it to OPTA. Remember that 2 programs are used, one for the M4 core and another for the M7 core.

ATTENTION: It is necessary to separate the memories and cores in the "Tools" tab of the IDE, as described below:

For programming with the name OPTA-PROG-PROCESSOR-INTERACTION-M7, follow these settings:

Tools > Flash split > 1MB M7 + 1MB M4

Tools > Target core > Main Core

After completing the setup, you can upload the programming to the product.

For programming with the name OPTA-PROG-PROCESSOR-INTERACTION-M4, follow these settings:

Tools > Flash split > 1MB M7 + 1MB M4

Tools > Target core > M4 Coprocessor

After completing the setup, you can send the schedule to the product.

5. Tests and Usage Examples

M7 Programming:

How it works:

1. **Initialization:** Sets the user button as input, starts RPC communication and initializes the M4 core.
2. **Main loop:**
 - Reads the user button state
 - If the button is pressed (LOW), calls the function `activateOutput()` on the M4 via RPC
 - If the button is not pressed, calls `deactivateOutput()` on the M4

- Adds a small delay to avoid excessive processing

Full System Flow

1. M7 constantly monitors the user's button
2. When the button is pressed:
 - M7 calls `RPC.call("enableOutput")`
 - M4 receives the call and executes `enableOutput()`, turning on D0 and LED_D0
3. When the button is released:
 - M7 calls `RPC.call("disableOutput")`
 - M4 receives the call and executes `disableOutput()`, turning off D0 and LED_D0

M4 Programming:

Operation:

1. **Initialization:** Configures pin D0 and LED_D0 as outputs and starts RPC communication with the M7.
2. **Registered functions:**
 - `enableOutput()` : Turns on pin D0 and LED_D0
 - `disableOutput()` : Turns off pin D0 and LED_D0
3. **Communication:** Functions are registered via `RPC.bind()` so that they can be called remotely by the M7.
4. **Loop:** Empty, since all actions are triggered by external RPC calls.