Desenvolv. de Sistemas Embarcados em Tempo Real

Prof. Hermano Cabral

Departamento de Eletrônica e Sistemas — UFPE

1 de agosto de 2024

Plano de Aula

Tema central

- Chibios — Hardware Abstraction Layer

Plano de Aula

Tema central

- Chibios — Hardware Abstraction Layer

Objetivos

- Conhecer as características do módulo de temporizadores do HAL

HAL - módulo de temporizadores

Introdução

- Este módulo implementa um driver para um temporizador de propósito geral.

HAL - módulo de temporizadores

Introdução

- Este módulo implementa um driver para um temporizador de propósito geral.
- O temporizador pode ser programado para disparar callbacks após um período especificado ou de forma periódica.

HAL - módulo de temporizadores

Arquivos

- Este módulo é implementado através dos seguintes arquivos dentro do diretório os/hal:
  - `include/hal_gpt.h` e `src/hal_gpt.c`, para a interface de alto nível do HAL
  - `ports/AVR/hal_gpt_lld.he` e `ports/AVR/hal_gpt_lld.c`, para a implementação de baixo nível em hardware

Arquivos de configuração

- Para se usar o driver GPT é preciso configurar os arquivos mcuconf.h e halconf.h.

Arquivos de configuração

- Para se usar o driver GPT é preciso configurar os arquivos mcuconf.h e halconf.h.
- No arquivo mcuconf.h, devemos mudar as linhas referentes ao temporizador que queremos usar:
  - `#define AVR_GPT_USE_TIM1 TRUE`
  - etc

Arquivos de configuração

- Para se usar o driver GPT é preciso configurar os arquivos mcuconf.h e halconf.h.
- No arquivo mcuconf.h, devemos mudar as linhas referentes ao temporizador que queremos usar:
  - `#define AVR_GPT_USE_TIM1 TRUE`
  - etc
- No arquivo halconf.h devemos selecionar a funcionalidade de GPT:
  - `#define HAL_USE_GPT TRUE`

HAL - módulo de temporizadores

```
typedef uint32_t gptfreq_t;
typedef void (*gptcallback_t)(GPTDriver *gptp);

typedef struct {
    gptfreq_t                   frequency;
    gptcallback_t               callback;
} GPTConfig;
```

Configuração

- A configuração deste módulo é feita através da estrutura GPTConfig e consiste em 2 coisas:
  - frequency – especificar a frequência do clock do temporizador

HAL - módulo de temporizadores

```
typedef uint32_t gptfreq_t;
typedef void (*gptcallback_t)(GPTDriver *gptp);

typedef struct {
    gptfreq_t                   frequency;
    gptcallback_t               callback;
} GPTConfig;
```

Configuração

- A configuração deste módulo é feita através da estrutura GPTConfig e consiste em 2 coisas:
  - frequency – especificar a frequência do clock do temporizador
  - callback – especificar a função de callback a ser chamada quando o timer expirar

Configuração

- Para a plataforma AVR, a frequência só é exata para alguns valores devido ao pequeno número de possibilidades para o divisor de frequência.

Configuração

- Para a plataforma AVR, a frequência só é exata para alguns valores devido ao pequeno número de possibilidades para o divisor de frequência.
- Para uma frequência do clock principal de 16 MHz, os valores exatos são 16 MHz, 2 MHz, 250 kHz, 62.5 kHz e 15625 Hz.

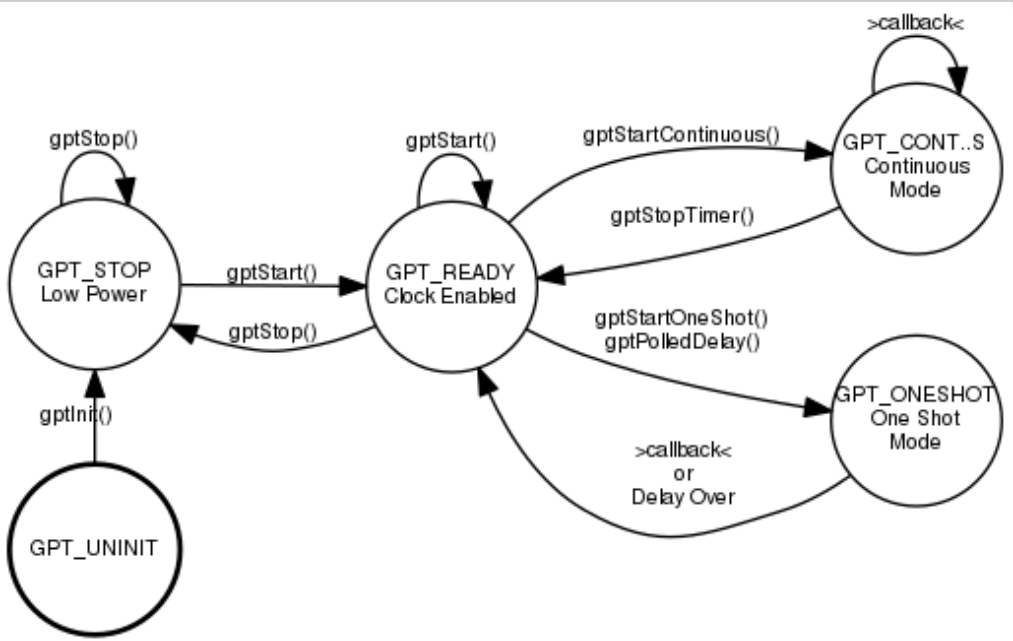
Configuração

- Para a plataforma AVR, a frequência só é exata para alguns valores devido ao pequeno número de possibilidades para o divisor de frequência.
- Para uma frequência do clock principal de 16 MHz, os valores exatos são 16 MHz, 2 MHz, 250 kHz, 62.5 kHz e 15625 Hz.
- Valores diferentes destes serão implementados de forma não-exata, embora com boa aproximação.
  - A aproximação piora à medida que a frequência cresce.

Configuração

- Para a plataforma AVR, a frequência só é exata para alguns valores devido ao pequeno número de possibilidades para o divisor de frequência.
- Para uma frequência do clock principal de 16 MHz, os valores exatos são 16 MHz, 2 MHz, 250 kHz, 62.5 kHz e 15625 Hz.
- Valores diferentes destes serão implementados de forma não-exata, embora com boa aproximação.
  - A aproximação piora à medida que a frequência cresce.
- Para todas as plataformas, a frequência mínima é limitada.

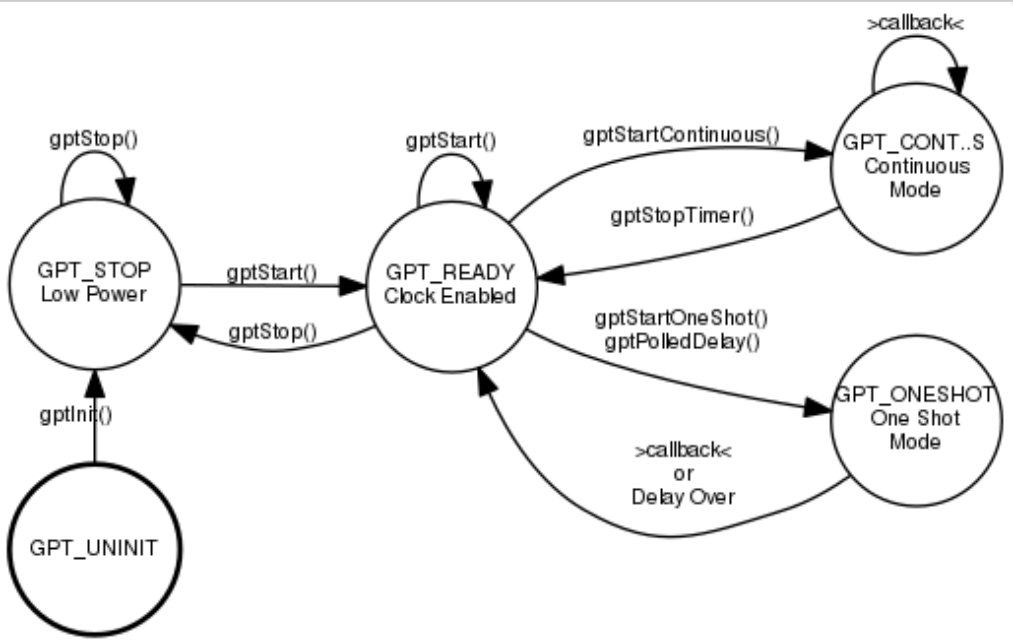
HAL - módulo de temporizadores



Máquina de estados

- Os principais estados da máquina de estados deste módulo são:
  - GPT_STOP – o clock do temporizador está parado

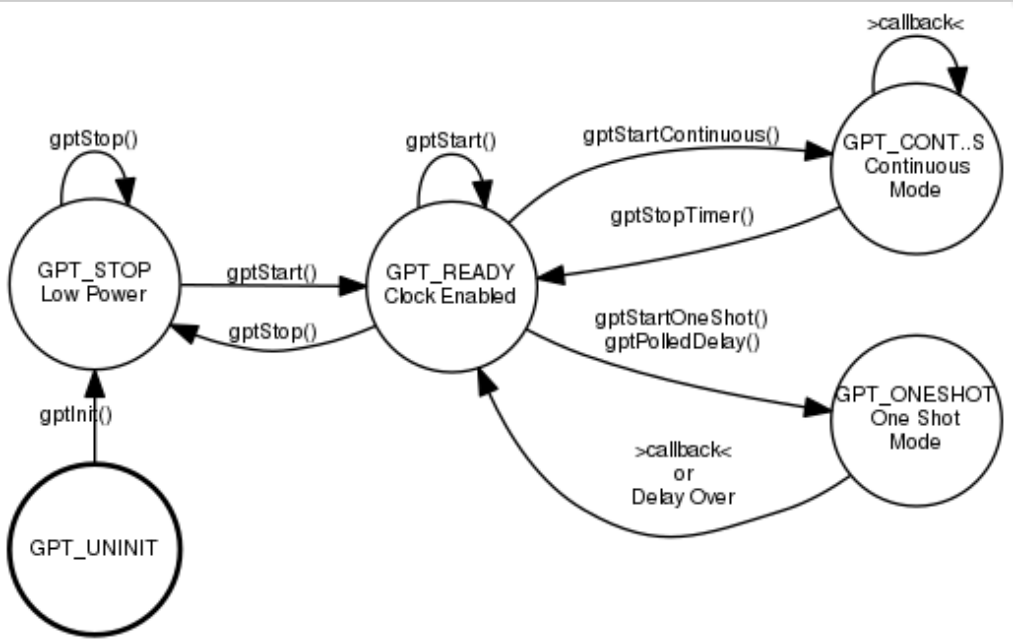
HAL - módulo de temporizadores



Máquina de estados

- Os principais estados da máquina de estados deste módulo são:
  - GPT_STOP – o clock do temporizador está parado
  - GPT_READY - o clock está habilitado mas não existe GPT configurado

HAL - módulo de temporizadores



Máquina de estados

- Os principais estados da máquina de estados deste módulo são:
  - GPT_STOP – o clock do temporizador está parado
  - GPT_READY - o clock está habilitado mas não existe GPT configurado
  - GPT_CONT_S e GPT_ONESHOT – GPT configurado e contando

HAL - módulo de temporizadores

Operação

- Para usarmos o temporizador seguimos o seguinte procedimento:
  - Criar uma variável para configuração do driver e inicializá-la

HAL - módulo de temporizadores

Operação

- Para usarmos o temporizador seguimos o seguinte procedimento:
  - Criar uma variável para configuração do driver e inicializá-la
  - Inicializar o driver com a configuração escolhida

Operação

- Para usarmos o temporizador seguimos o seguinte procedimento:
  - Criar uma variável para configuração do driver e inicializá-la
  - Inicializar o driver com a configuração escolhida
  - Habilitar o GPT chamando gptStartContinuous(), gptStartOneShot() ou gptPolledDelay()

Operação

- A habilitação do GPT pode-se dar por 3 meios:
  - Inicia o GPT de forma periódica

Operação

- A habilitação do GPT pode-se dar por 3 meios:
  - Inicia o GPT de forma periódica
  - Inicia o GPT de forma disparo único

Operação

- A habilitação do GPT pode-se dar por 3 meios:
  - Inicia o GPT de forma periódica
  - Inicia o GPT de forma disparo único
  - Inicia o GPT de forma disparo único e espera o fim da contagem

Principais funções

- As principais funções são:
  - gptStart() – configura o temporizador e prepara o GPT para poder ser usado

Principais funções

- As principais funções são:
  - gptStart() – configura o temporizador e prepara o GPT para poder ser usado
  - gptStop() – para o temporizador

Principais funções

- As principais funções são:
  - gptStart() – configura o temporizador e prepara o GPT para poder ser usado
  - gptStop() – para o temporizador
  - gptStartContinuous() e gptStartOneShot() – inicia a contagem do GPT de acordo com o período passado como parâmetro e configura o callback para ser chamado ao final da contagem

Principais funções

- As principais funções são:
  - gptStart() – configura o temporizador e prepara o GPT para poder ser usado
  - gptStop() – para o temporizador
  - gptStartContinuous() e gptStartOneShot() – inicia a contagem do GPT de acordo com o período passado como parâmetro e configura o callback para ser chamado ao final da contagem
  - gptStopTimer() – para o GPT

Principais funções

- As principais funções são:
  - gptStart() – configura o temporizador e prepara o GPT para poder ser usado
  - gptStop() – para o temporizador
  - gptStartContinuous() e gptStartOneShot() – inicia a contagem do GPT de acordo com o período passado como parâmetro e configura o callback para ser chamado ao final da contagem
  - gptStopTimer() – para o GPT
  - gptPolledDelay() – inicia a contagem do GPT de acordo com o período passado como parâmetro e espera o fim da contagem

HAL - módulo de temporizadores

```
typedef uint16_t gptcnt_t;

void gptInit(void);
void gptStart(GPTDriver *gptp, const GPTConfig *config);
void gptStop(GPTDriver *gptp);
void gptStartContinuous(GPTDriver *gptp, gptcnt_t interval);
void gptStartOneShot(GPTDriver *gptp, gptcnt_t interval);
void gptStopTimer(GPTDriver *gptp);
void gptPolledDelay(GPTDriver *gptp, gptcnt_t interval);
void gptChangeInterval(GPTDriver *gptp, gptcnt_t interval);
```

Principais funções

- As assinaturas das funções estão mostradas acima.

HAL - módulo de temporizadores

Principais funções

- Os drivers são declarados como GPTD1, GPTD2, GPTD3, etc.

HAL - módulo de temporizadores

Principais funções

- Os drivers são declarados como GPTD1, GPTD2, GPTD3, etc.
- Os valores para os intervalos de tempo são medidos em pulsos de clock.

HAL - módulo de temporizadores

```
void gpt_cb(GPTDriver* gptd) {
    (void) gptd;
    palTogglePad(LED_PORT, LED_PIN);
}

int main(void) {
    GPTConfig driver_config = {.frequency = GPT_FREQUENCIA,
                               .callback = gpt_cb
    };

    halInit();
    chSysInit();

    palSetPadMode(LED_PORT, LED_PIN, PAL_MODE_OUTPUT_PUSHPULL);
    palClearPad(LED_PORT, LED_PIN);

    gptStart(&GPTD1, &driver_config);
    gptStartContinuous(&GPTD1, LED_PERIODO);

    while (1) {}
}
```

Principais funções

- Como exemplo, podemos usar a funcionalidade do GPT para implementar os intervalos de tempo da geração do sinal PWM pela CPU.