Insper

Computação embarcada 2023-1

LAB 4 – RTOS – ADC

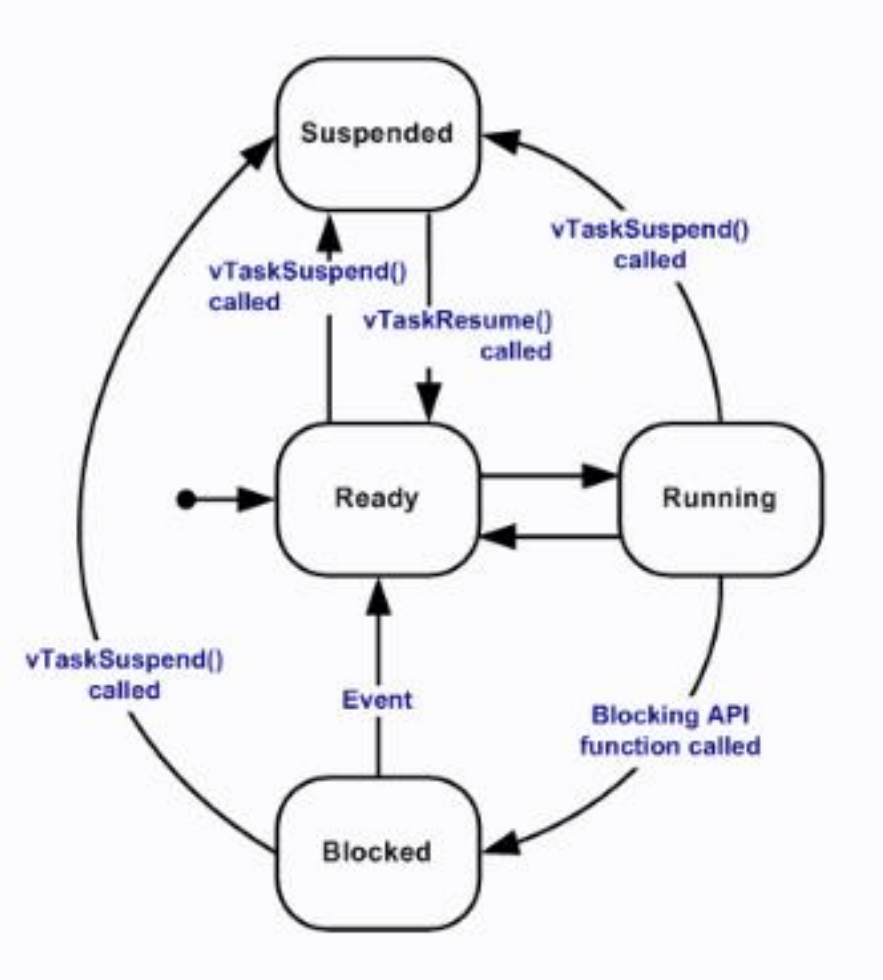
TASKS

Rodam em loops infinitos e nunca retornam

```
static void task_led(void *pvParameters) {
//o atraso para essa funcao ser chamada apos um ajus
  LED_init(1);

  uint32_t msg = 0;
  uint32_t delayMs = 2000;

  /* tarefas de um RTOS não devem retornar */
  for (;;) {
                //rotina
          }
  }
```

TASKS

Podem estar em diferentes estados



TASKS

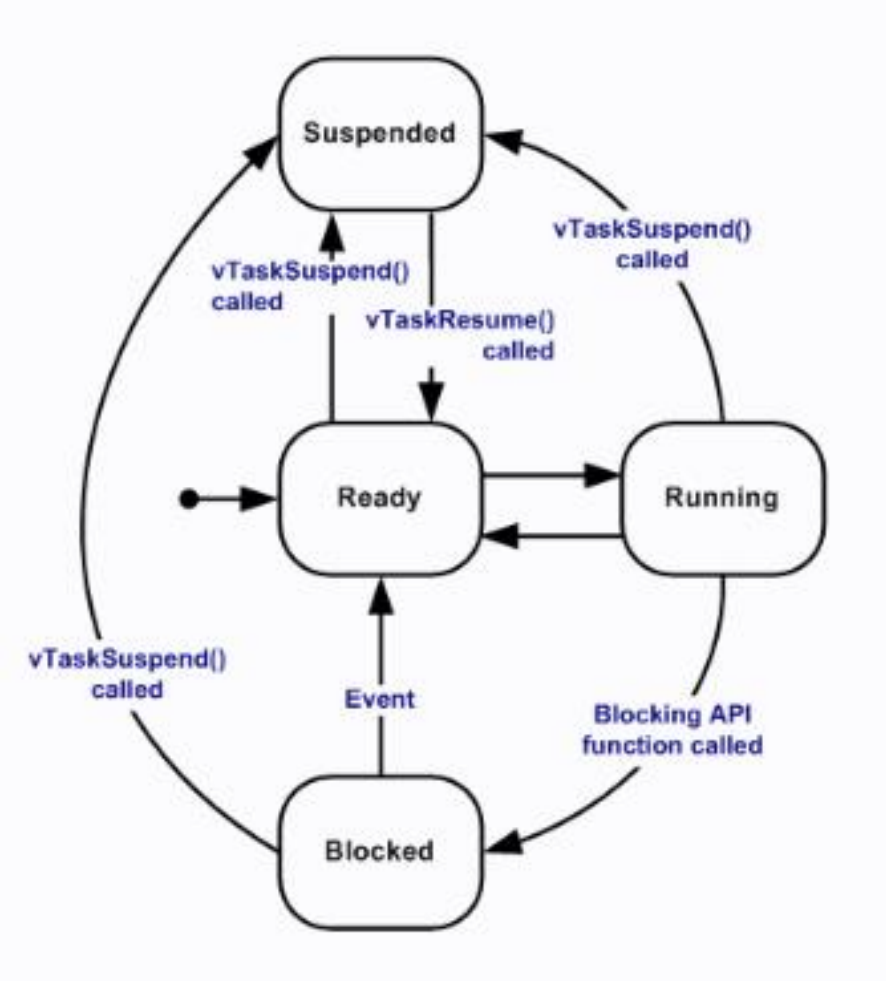
Possuem stacks e prioridades

```
#define TASK_LED_STACK_SIZE (4096 / sizeof(portSTACK_TYPE))
#define TASK_LED_STACK_PRIORITY (tskIDLE_PRIORITY)
#define TASK_BUT_STACK_SIZE (4096 / sizeof(portSTACK_TYPE))
#define TASK_BUT_STACK_PRIORITY (tskIDLE_PRIORITY)
```

Possuem queues

```
if(xQueueReceive(xQueueLedFreq_2,&msg,(TickType_t) 0))
{

  /* envia nova frequencia para a task_led */
  xQueueSend(xQueueLedFreq, (void *)&delayTicks, 10);
```

PERIODICIDADE DAS TASKS



Tasks são executadas durante um certo intervalo de tempo, DEPOIS SUSPENSAS, liberando o processador!

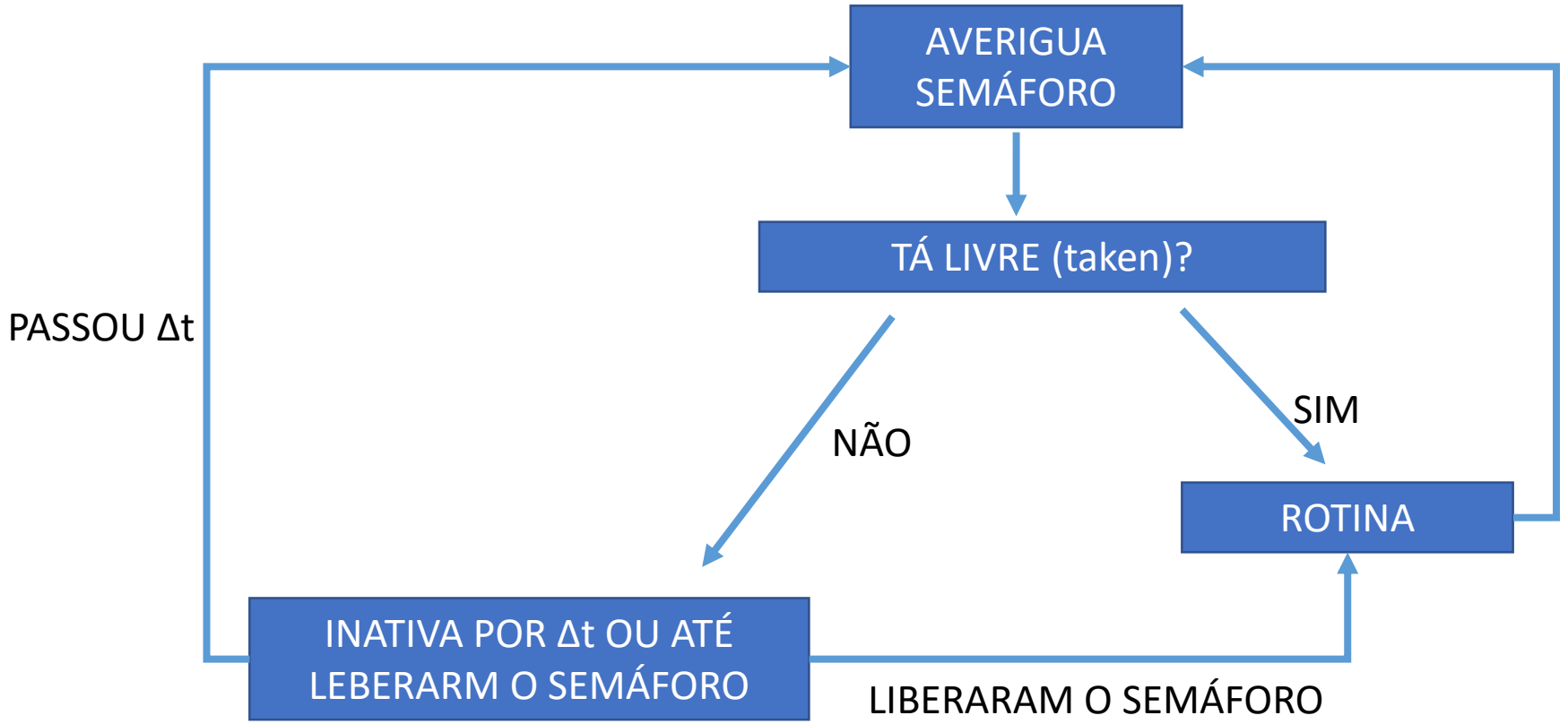
VtASKdELAY(TICKS) ou xSemaphoreTake

```
/* suspende por delayMs */
vTaskDelay(delayTicks / portTICK_PERIOD_MS);
```

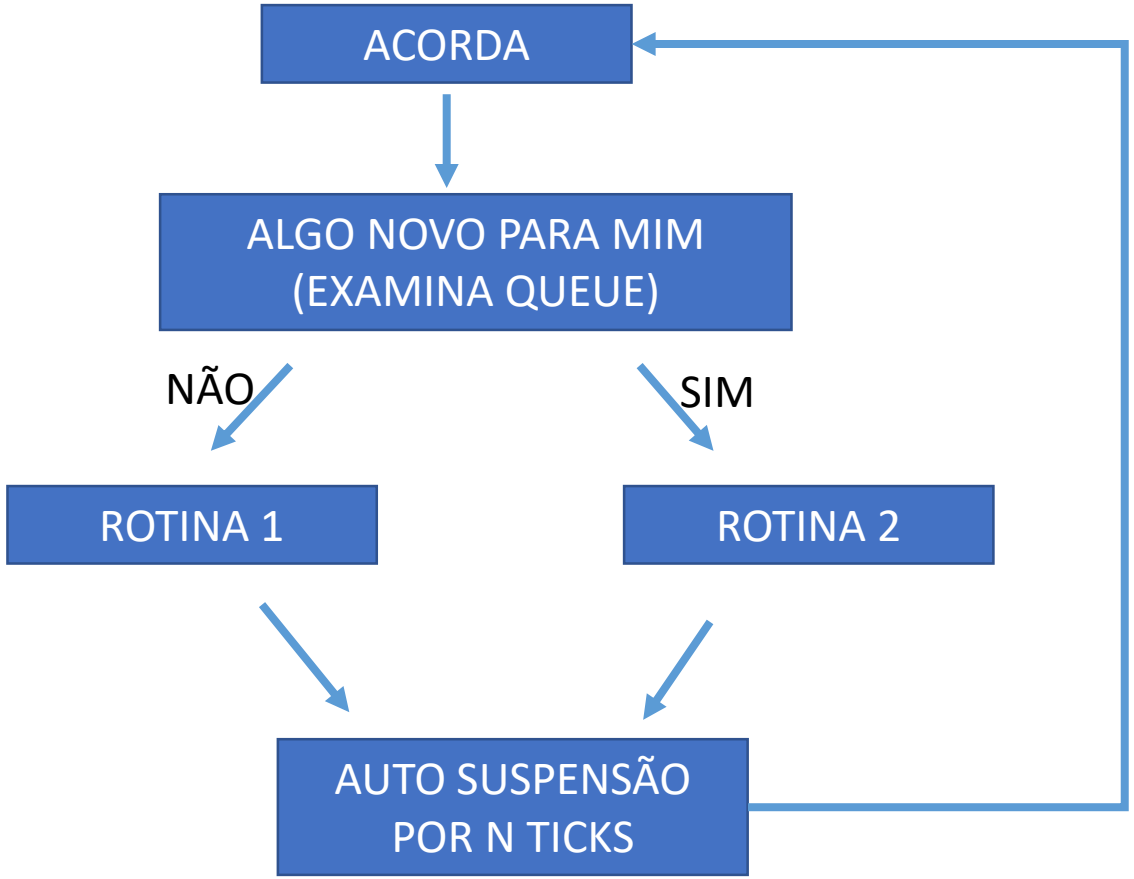
```
if (xQueueReceive(xQueueADC, &(adc), 1000)) {
```

```
if (xSemaphoreTake(xSemaphoreBut,1000)) {
```

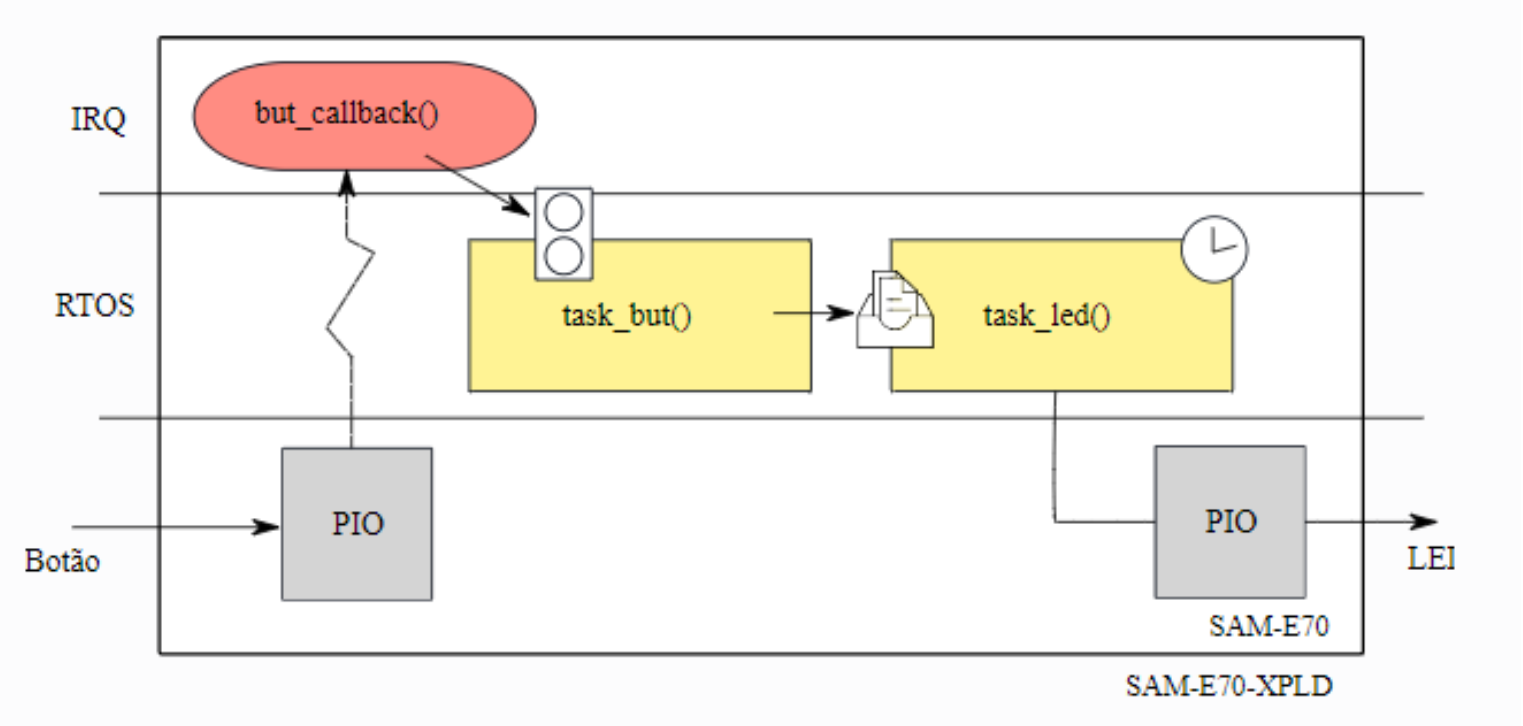
FORMAS DE VIDA DE UMA TASK



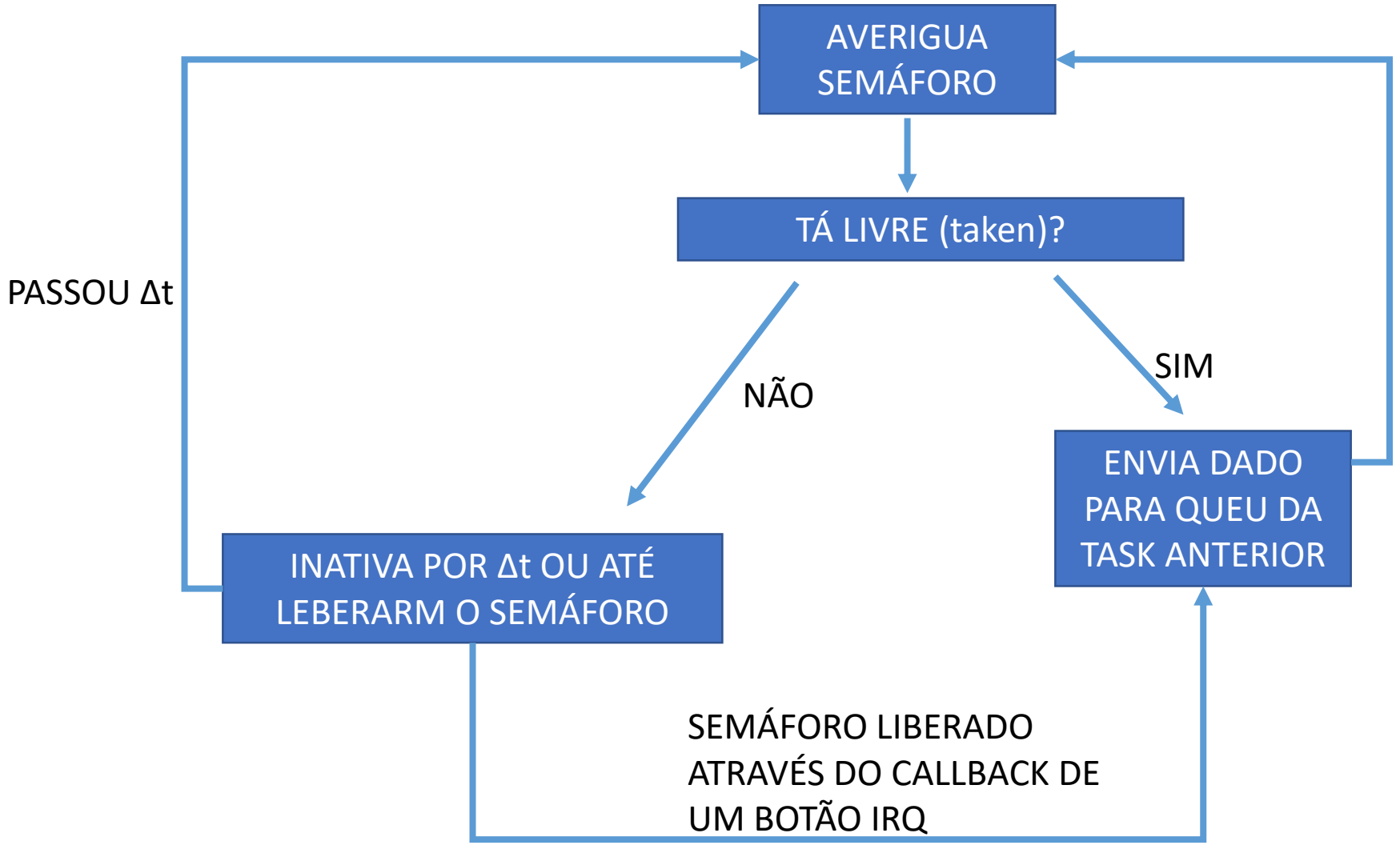
FORMAS DE VIDA DE UMA TASK



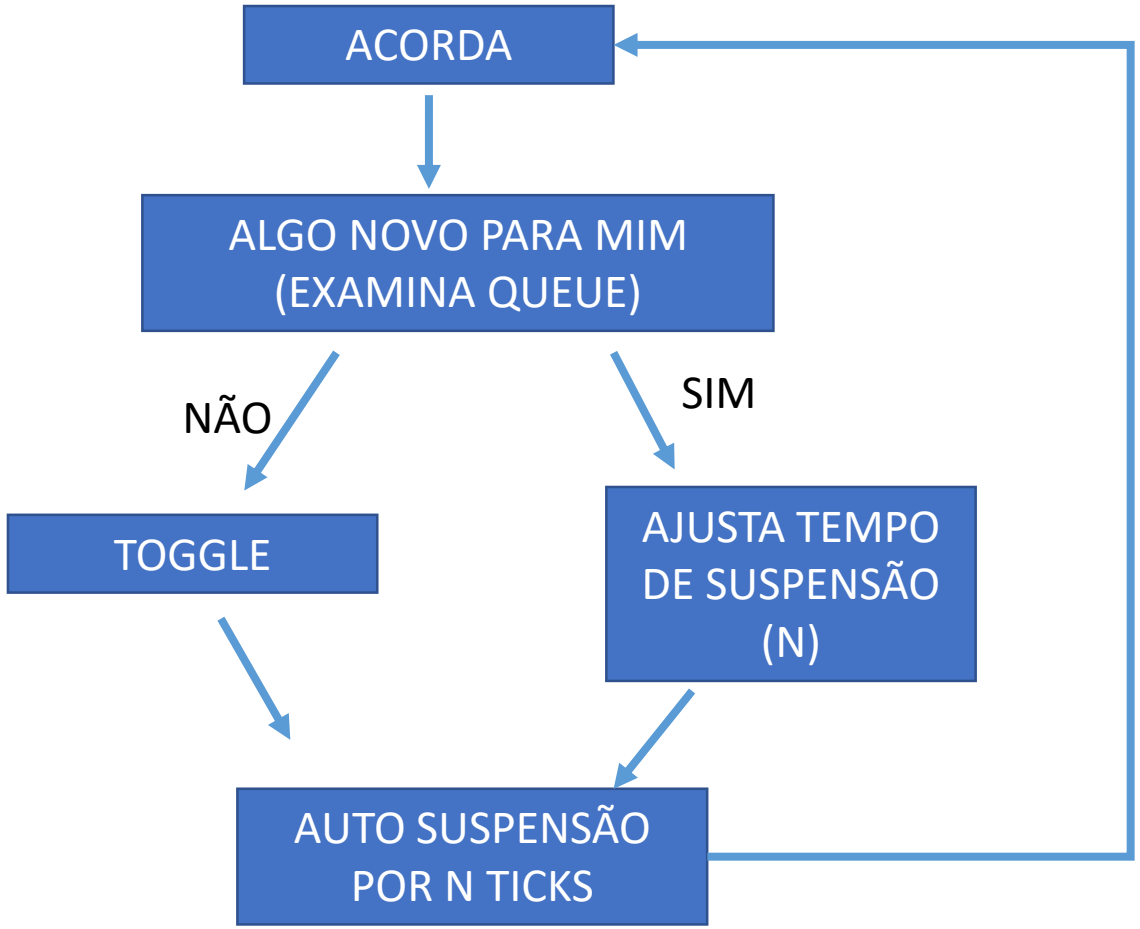
INTERRUPT + SEMAPHORE + TASKS – RECURSOS DO RTOS



FORMAS DE VIDA DE UMA TASK

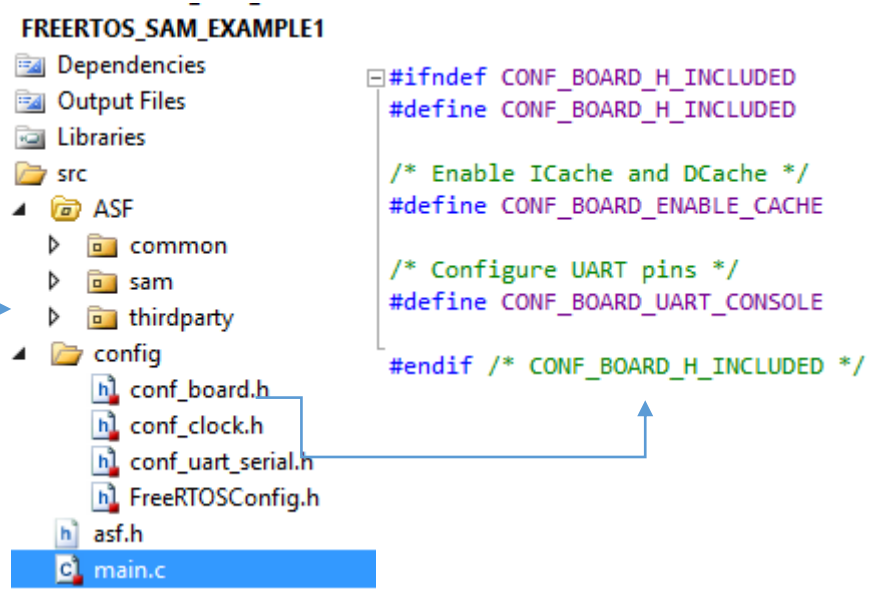


FORMAS DE VIDA DA TASK 1



MONITOR SERIAL

1) colocar a biblioteca board.h em config e adicionar os defines da figura:



2) copiar do projeto 4 a funcao ***static void configure_console(void)***; (não apague o conteúdo relativo a outras coisas, OLED por exemplo).

3) Chamar a função no main: ***configure_console();***

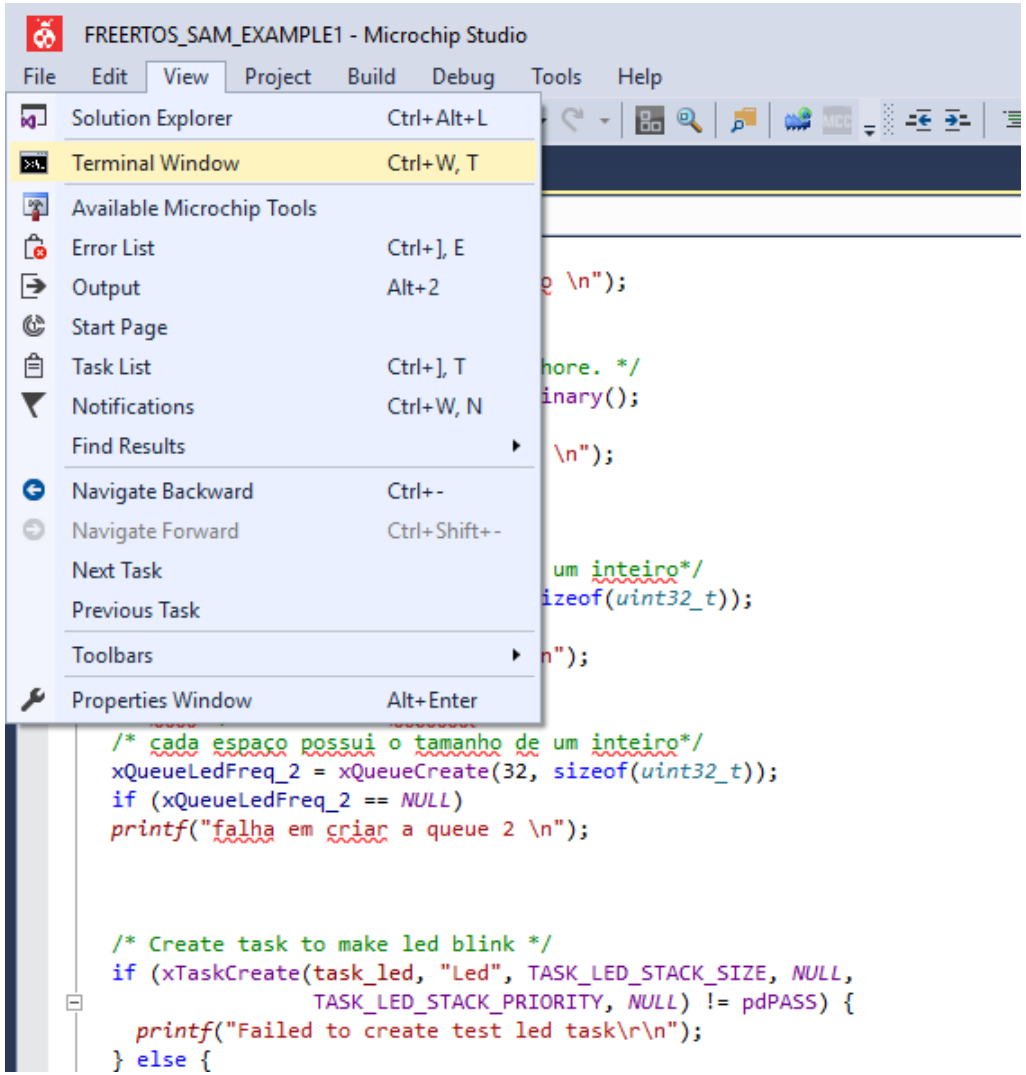
4) No ASF adicionar o ***standard serial***

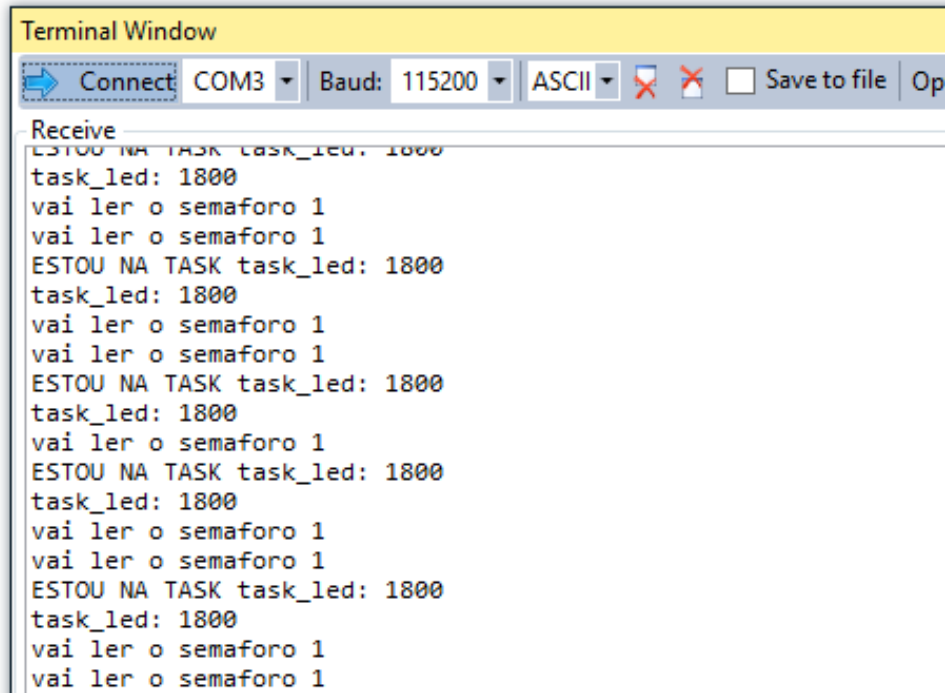
5) Copiar o conteúdo do conf_uart_serial do exemplo 4.

6) Instalar o plugin do Terminal Windows no Microship Studio

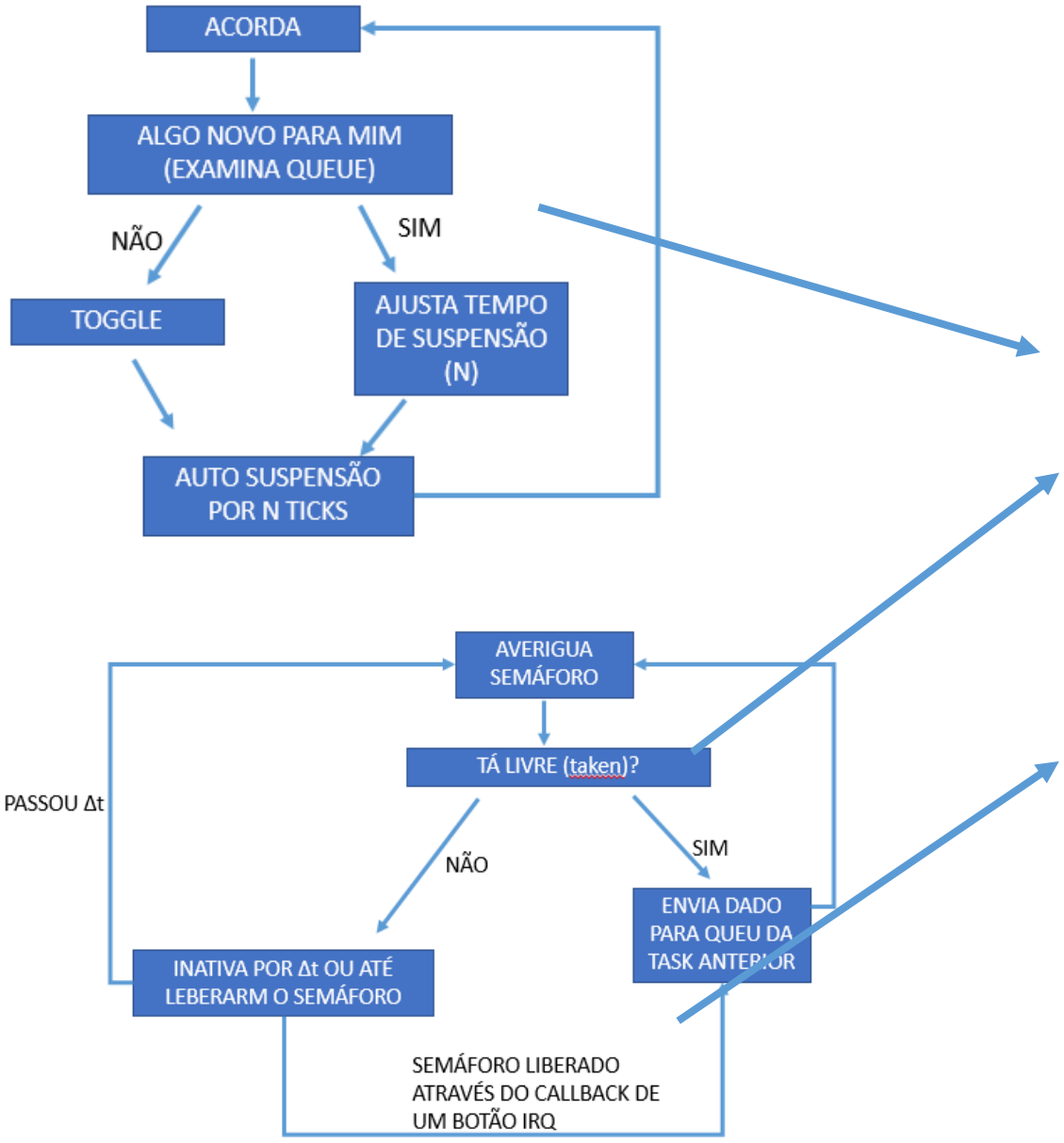
```
/**
 * \brief Configure the console UART.
 */
static void configure_console(void) {
  const usart_serial_options_t uart_serial_options = {
      .baudrate = CONF_UART_BAUDRATE,
      .charlength = CONF_UART_CHAR_LENGTH,
      .paritytype = CONF_UART_PARITY,
      .stopbits = CONF_UART_STOP_BITS,
  };

  /* Configure console UART. */
  stdio_serial_init(CONF_UART, &uart_serial_options);

  /* Specify that stdout should not be buffered. */
  setbuf(stdout, NULL);
}
```

MONITOR SERIAL






```
if (xQueueLedFreq == NULL)
  printf("falha em criar a queue \n");
```



Terminal Window

Connect COM3 Baud: 115200 ASCII Save to file Options

Receive

```
 o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
ESTOU NA TASK task_led: 4200
task_led: 4200
vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
ESTOU NA TASK task_led: 4200
task_led: 4200
vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
ESTOU NA TASK task_led: 4200
task_led: 4200
vai ler o semaforo 1 com inatividade de 1000 ms
liberou o semafro ...task_but: 4100

vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
vai ler o semaforo 1 com inatividade de 1000 ms
ESTOU NA TASK task_led: 4200
task_led: 4100
```

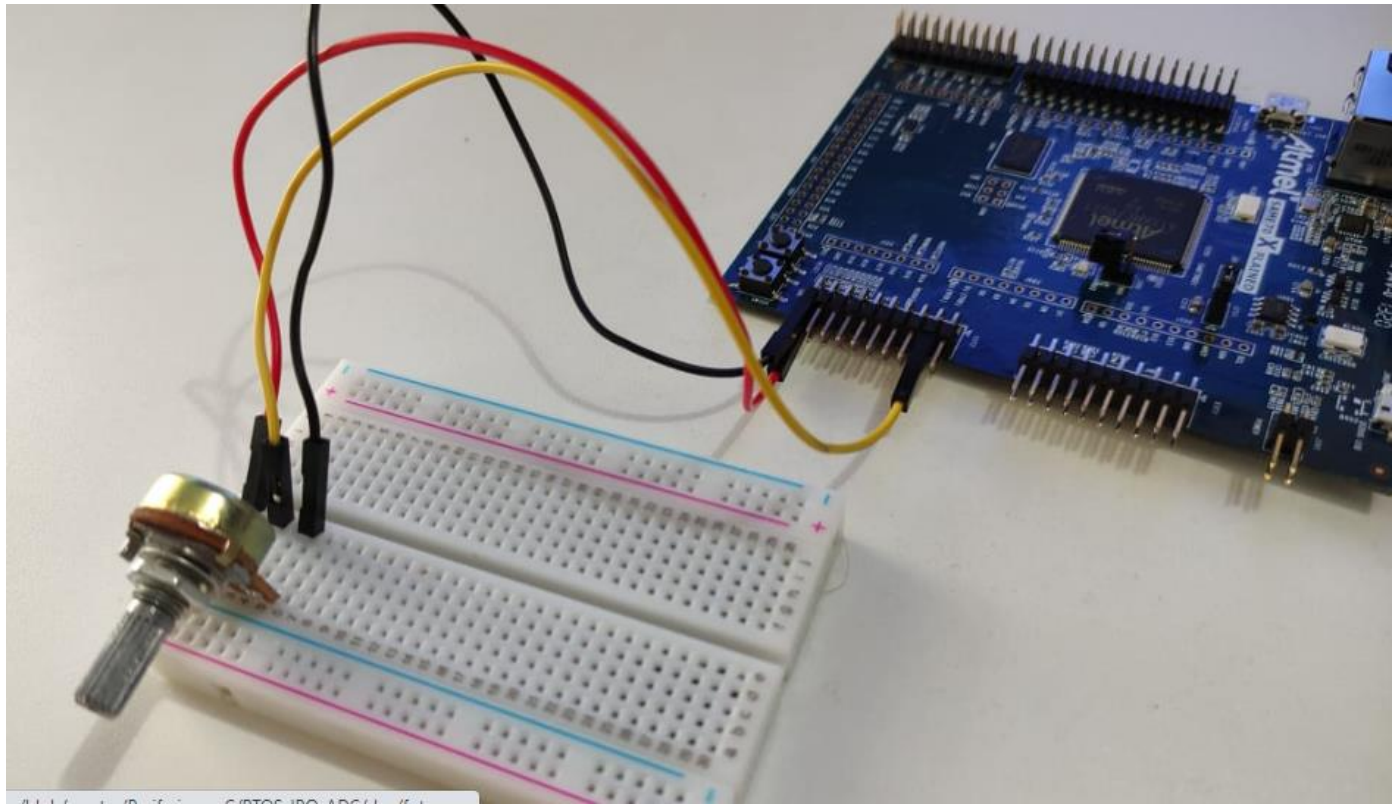
ADC

AFEC - PIN

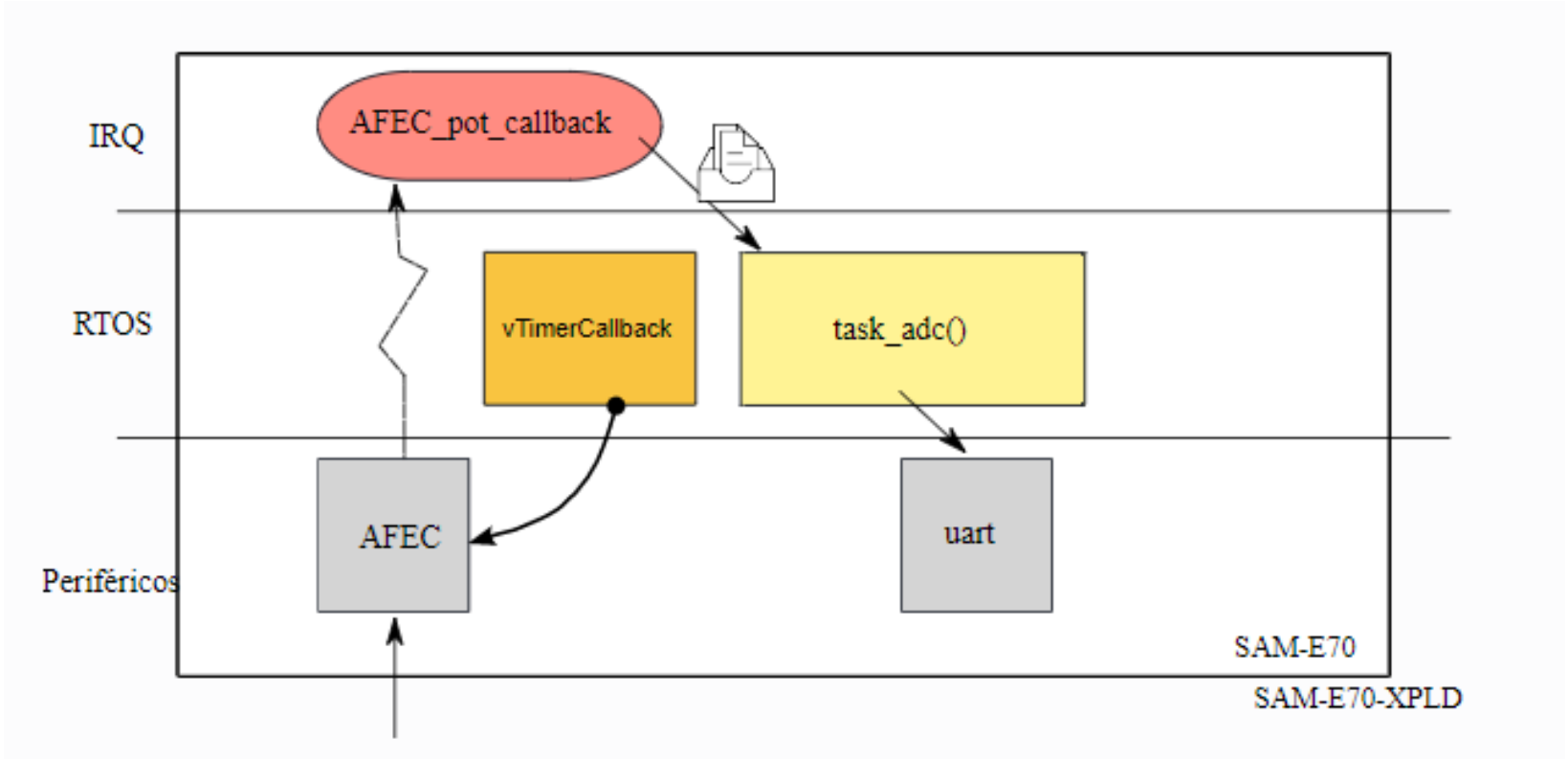
Esse exemplo efetua a leitura de um potenciômetro e imprime o valor lido na serial.

- Periféricos:
  - AFEC0 - Analog Front-End Controller
  - USART1 (debug - para comunicação com o PC - `stdio`)
- Pinos:
  - EXT-2: pin 3
    - `PD30` : AFEC0
- Componentes:
  - SAME70-XPLD
  - 1x Potenciômetro 10k
  - Jumpers

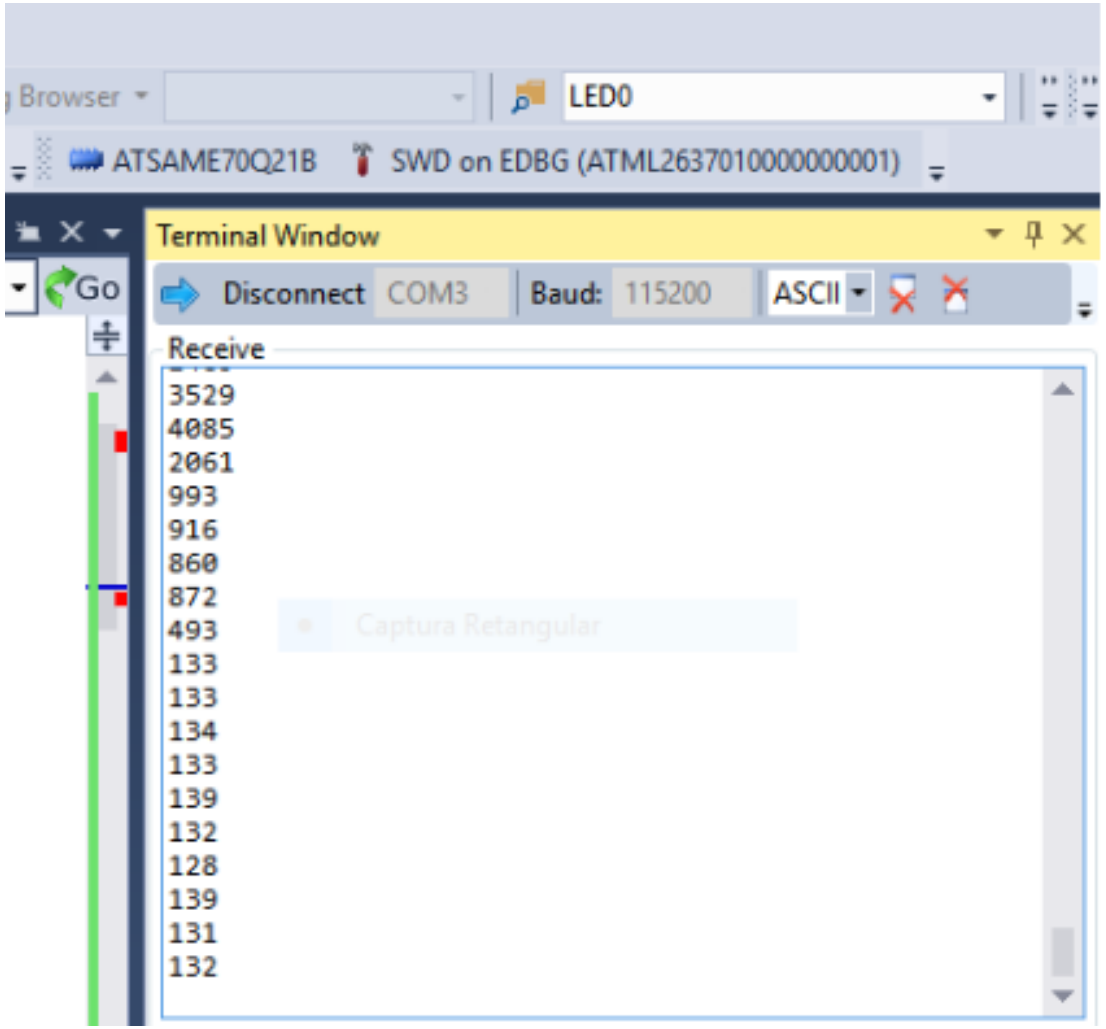
ADC



ADC


IRQ
RTOS
Periféricos
AFEC_pot_callback
vTimerCallback
task_adc()
AFEC
uart
SAM-E70
SAM-E70-XPLD

SAÍDA


Browser
LED0
ATSAME70Q21B
SWD on EDBG (ATML2637010000000001)
Terminal Window
Go
Disconnect
COM3
Baud:
115200
ASCII
Receive
3529
4085
2061
993
916
860
872
493
133
133
134
133
139
132
128
139
131
132