

CONTROLE DE TEMPERATURA UTILIZANDO SISTEMA OPERACIONAL DE TEMPO REAL

ARTHUR DAMASCENO

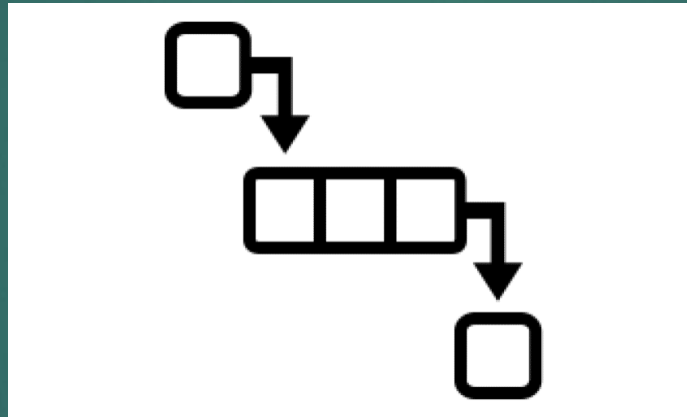
MATEUS PICCININ

Descrição do problema

- Planta: Resistor de Potência ligado a uma fonte 12V
- Sensoriamento: Termopar do Tipo K
- Microcontrolador: STM32F103C8T6
- Variável controlada: Razão Cíclica do MOSFET
- Variável de controle: Temperatura do resistor de potência

Recursos utilizados

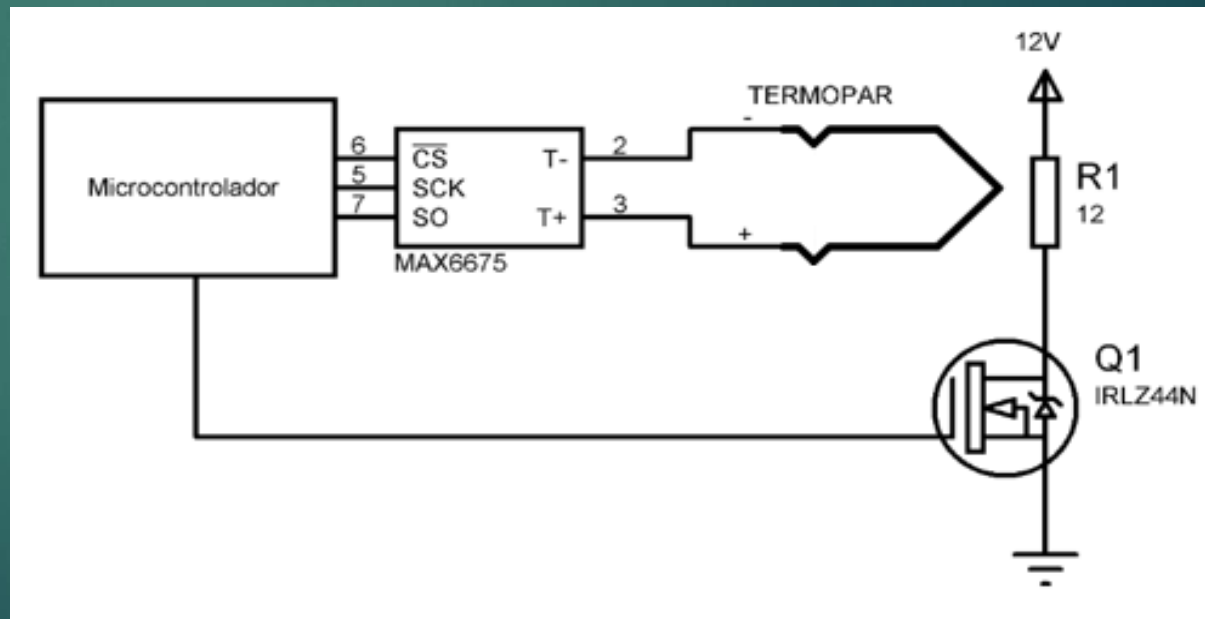
- ▶ Sincronização/Comunicação: Filas



- ▶ Temporização: Atraso

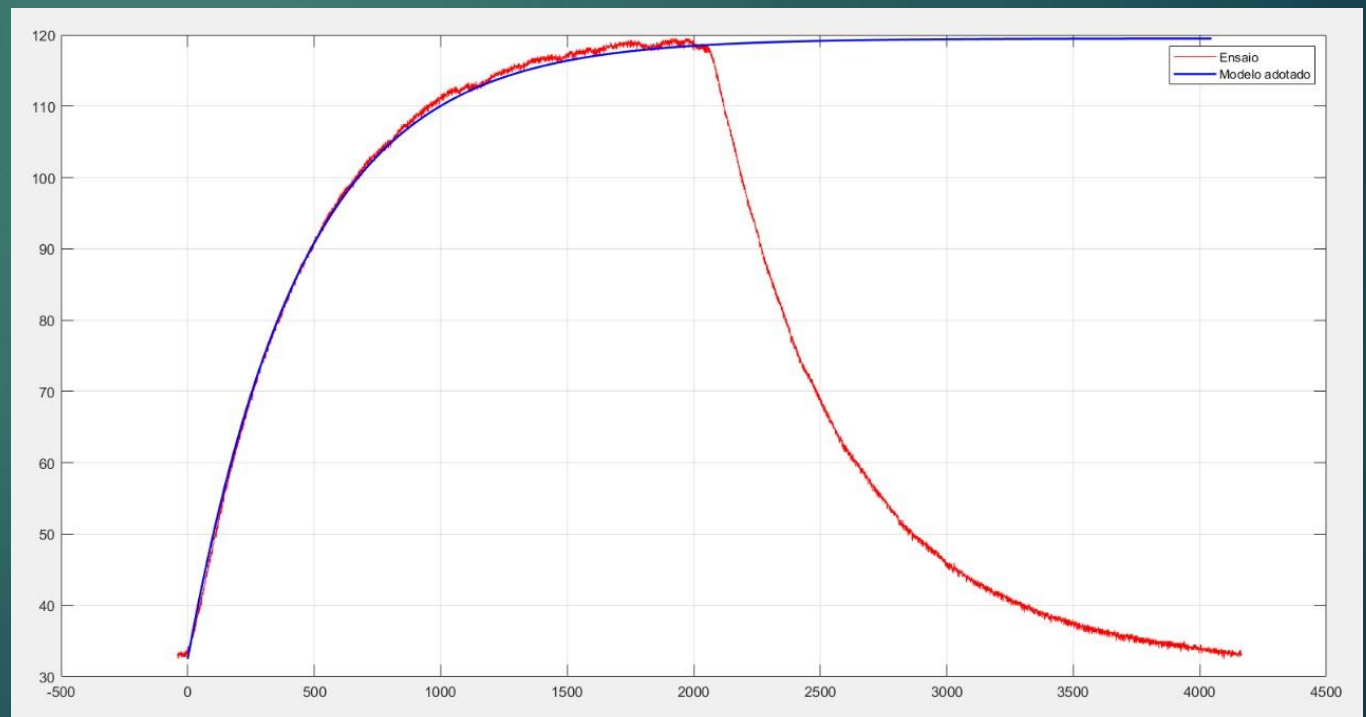
Circuito de Aquisição

- ▶ Termopar tipo K
- ▶ MAX6675
- ▶ Microcontrolador
- ▶ Razão Cíclica 100%



Função de Transferência

► $G(s) = \frac{87}{450.5s+1}$



Controlador e Filtro FIR

- ▶ Controlador Proporcional-Integral (PI)
- ▶ Controlador ON/OFF
- ▶ Implementação em Equação de Diferenças com Anti-Windup
- ▶ Filtro Digital FIR para leituras de temperatura

Sistema Operacional

- ▶ Criação das Tarefas na pilha do escalonador
 - ▶ Temp_Task
 - ▶ Control_Task
 - ▶ Filter_Task
 - ▶ Display_Task
- ▶ Criação de Filas
 - ▶ TempQueue
 - ▶ FilteredTempQueue

Temp_Task

- ▶ Ler temperatura
- ▶ Atraso de 200ms
- ▶ Prioridade 3
- ▶ Envio para fila TempQueue

```
void Temp_taskF(void *pvParameters) {
    while (1) {
        uint8_t tempdata[16];
        uint16_t temp16 = 0;

        /* bitbanging protocolo SPI */
        CSen
        for (int i = 0; i < 16; i++) {
            SCK_H
            tempdata[i] = HAL_GPIO_ReadPin(GPIOB, GPIO_PIN_4);
            SCK_L
        }
        CSdis

        /* Conversão temperatura */
        if (tempdata[13] == 0) {

            for (int n = 1; n < 13; n++) {
                temp16 += tempdata[n] * (2048 / (1 << (n - 1)));
            }

        }

        float temp = (float) temp16 / 4;

        /* Adiciona a fila tempQueue */
        if (xQueueSend(tempQueue, &temp, 10) == pdPASS) {
        }

        /* Atraso para definição do período da tarefa */
        vTaskDelay(200); /*5Hz frequency*/
    }
}
```


Filter_Task

- ▶ Prioridade 2
- ▶ Recebe valor da fila TempQueue
- ▶ Filtra a variável
- ▶ Após 5 valores envia para fila FilteredTempQueue

```
void Filter_taskF(void *pvParameters) {
    uint8_t aux = 0, aux2=1;
    while (1) {
        float rx_temp;
        /* Recebe da fila filteredTempQueue */
        if (xQueueReceive(tempQueue, &rx_temp, 10)) {
            aux++;
            /* Chamada do filtro FIR */
            filteredTemp = FIRFilter_Update(&tempFilter, rx_temp);

        }
        if(aux==4 && aux2==1){
            aux2=0;
            SEGGER_SYSVIEW_Start();
        }
        if (aux == 5) {

            aux = 0;
            /* Adiciona a fila filteredTempQueue */
            if (xQueueSend(filteredTempQueue, &filteredTemp, 10) == pdPASS) {
            }

        }

    }
}
```

Control_Task

- ▶ Prioridade 4
- ▶ Recebe valor de temperatura da fila FilteredTempQueue
- ▶ Lê referencia
- ▶ Calcula ação de controle
- ▶ Aciona atuador

```
void Control_taskF(void *pvParameters) {
    while (1) {
        float rx_filteredTemp;
        /* Recebe da fila filteredTempQueue */
        if (xQueueReceive(filteredTempQueue, &rx_filteredTemp, 10)) {
            /* Leitura da entrada analógica para calculo de referencia */
            HAL_ADC_PollForConversion(&hadc1, 10);
            ref = (float) HAL_ADC_GetValue(&hadc1) / 27.3; // leitura do potenciometro convertido em ref até 150°C

            /* Lei de controle */
            float u;
            static float up, uint;
            int flag_sat;
            float ek = ref - rx_filteredTemp;

            /* Controlador bang-bang ventoinha */
            if (ek < -15.0) {
                HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, GPIO_PIN_SET);
            } else {
                HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, GPIO_PIN_RESET);
            }

            /* Anti-windup integrador */
            if (!flag_sat) {
                uint = uint * p1 + r1 * ek;
            } else if (flag_sat) {
                uint = (uint * p1 + r1 * ek) * ksats;
            }

            /* Proporcional */
            up = k * ek;
        }
    }
}
```

Display_Task

- ▶ Prioridade 1
- ▶ Atraso de 1000ms
- ▶ Atualiza mostrador gráfico – UART

```
void Display_taskF(void *pvParameters) {
    while (1) {
        char str[100];
        /* Fim de comando definido pela API do display */
        uint8_t Cmd_End[3] = { 0xFF, 0xFF, 0xFF };

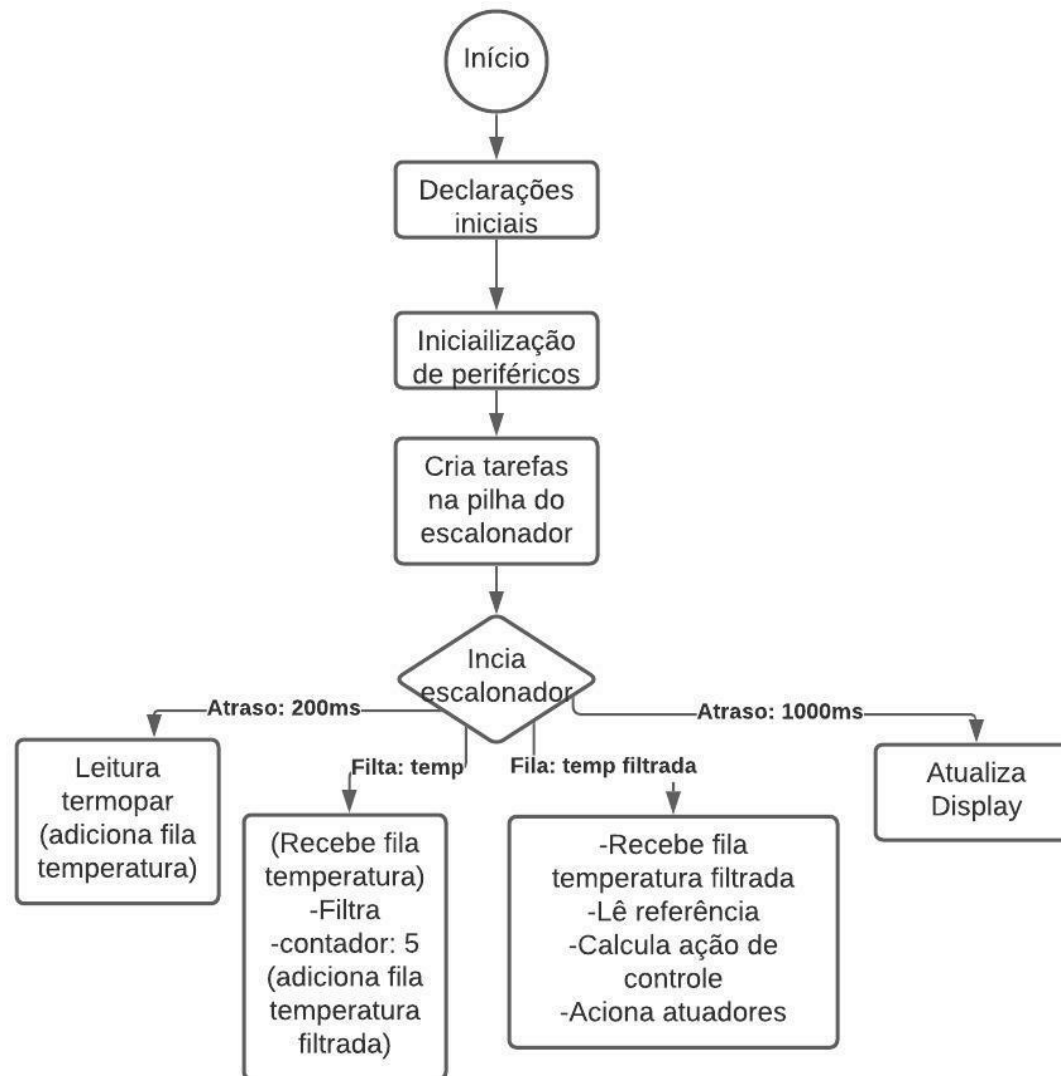
        /* Atualiza valor do setpoint */
        int32_t number = ref * 100;
        sprintf(str, "setPoint.val=%ld", number);
        HAL_UART_Transmit(&huart1, (uint8_t*) str, strlen(str), 10);
        HAL_UART_Transmit(&huart1, Cmd_End, 3, 10);

        /* Atualiza valor da variável de processo */
        number = filteredTemp * 100;
        sprintf(str, "filteredTemp.val=%ld", number);
        HAL_UART_Transmit(&huart1, (uint8_t*) str, strlen(str), 10);
        HAL_UART_Transmit(&huart1, Cmd_End, 3, 10);

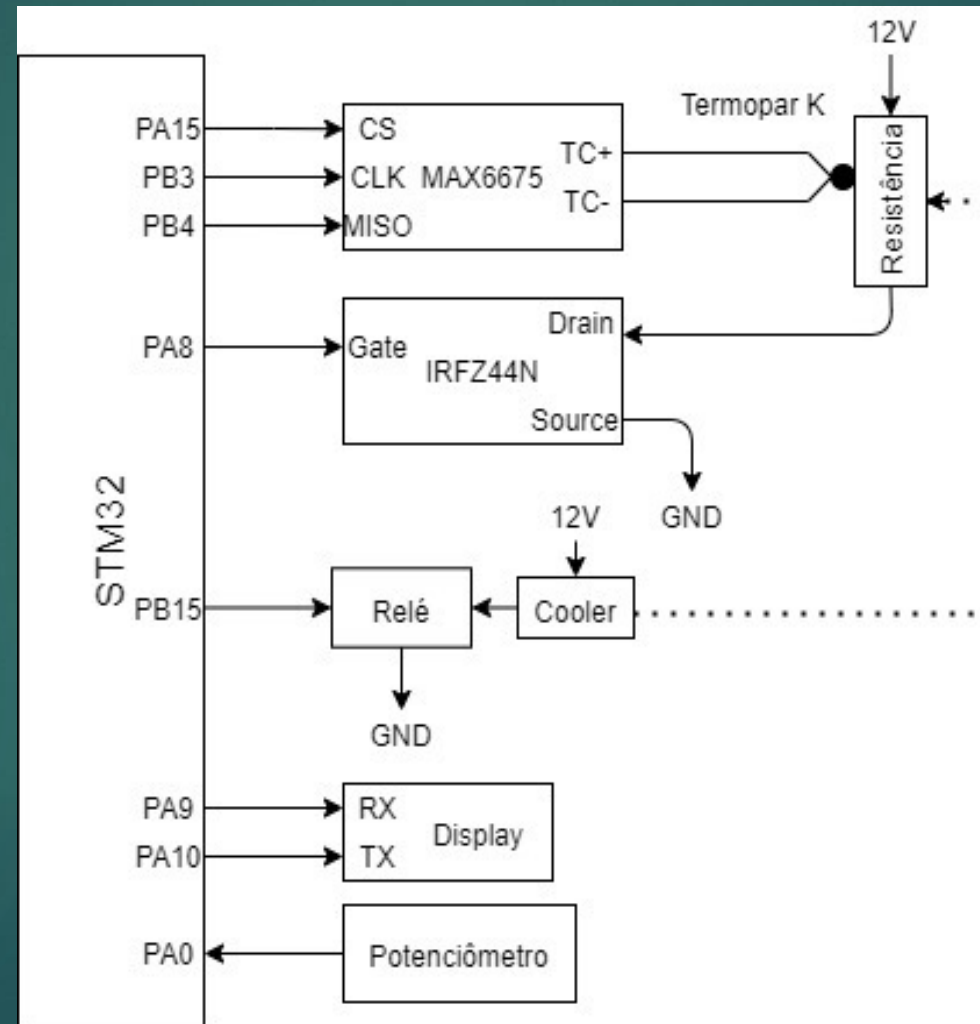
        /* Atualiza valor da variável manipulada */
        number = dutyCycle * 100;
        sprintf(str, "dutyCycle.val=%ld", number);
        HAL_UART_Transmit(&huart1, (uint8_t*) str, strlen(str), 10);
        HAL_UART_Transmit(&huart1, Cmd_End, 3, 10);

        /* Atraso para definição do período da tarefa */
        vTaskDelay(1000); /*1Hz frequency*/
    }
}
```

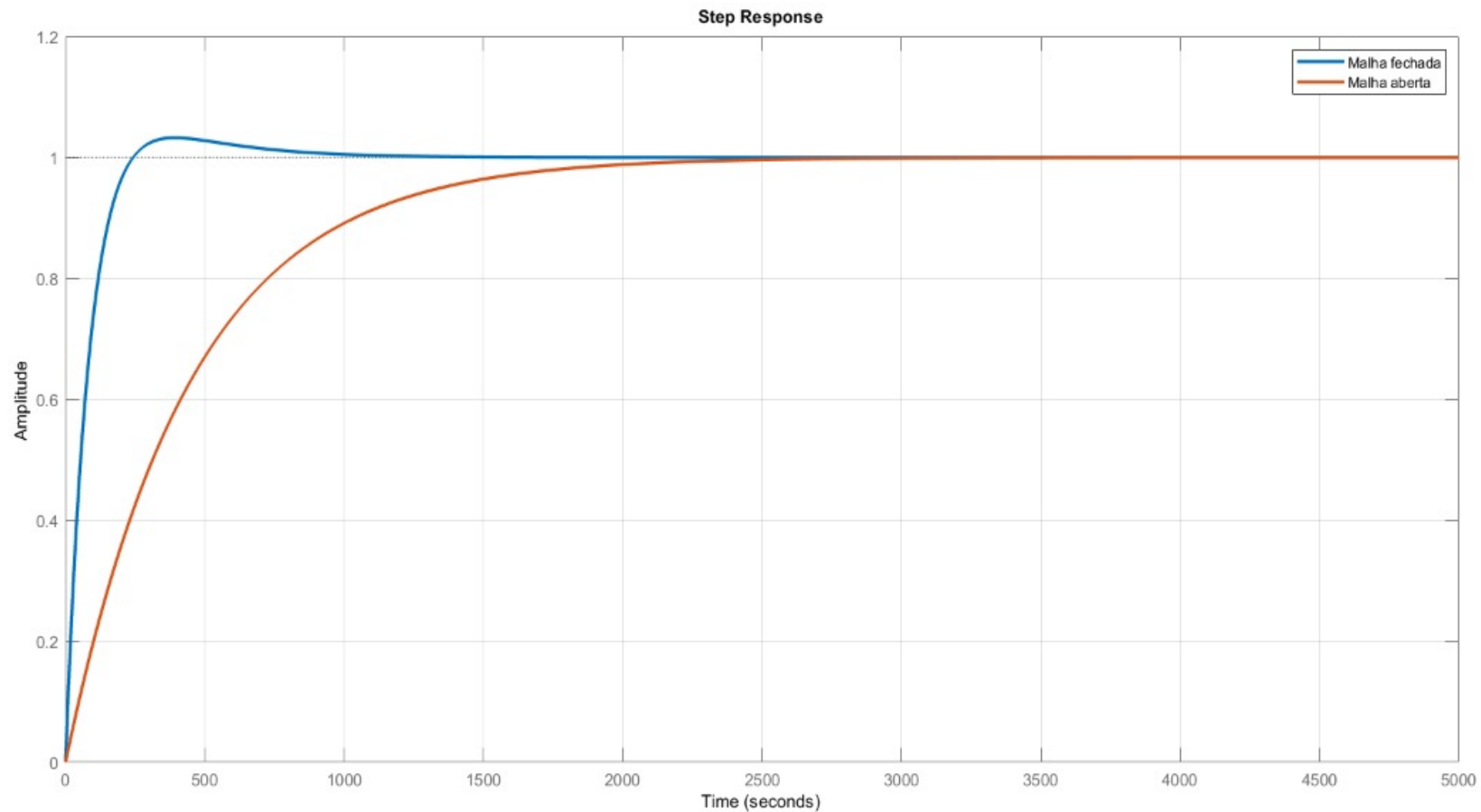
Fluxograma



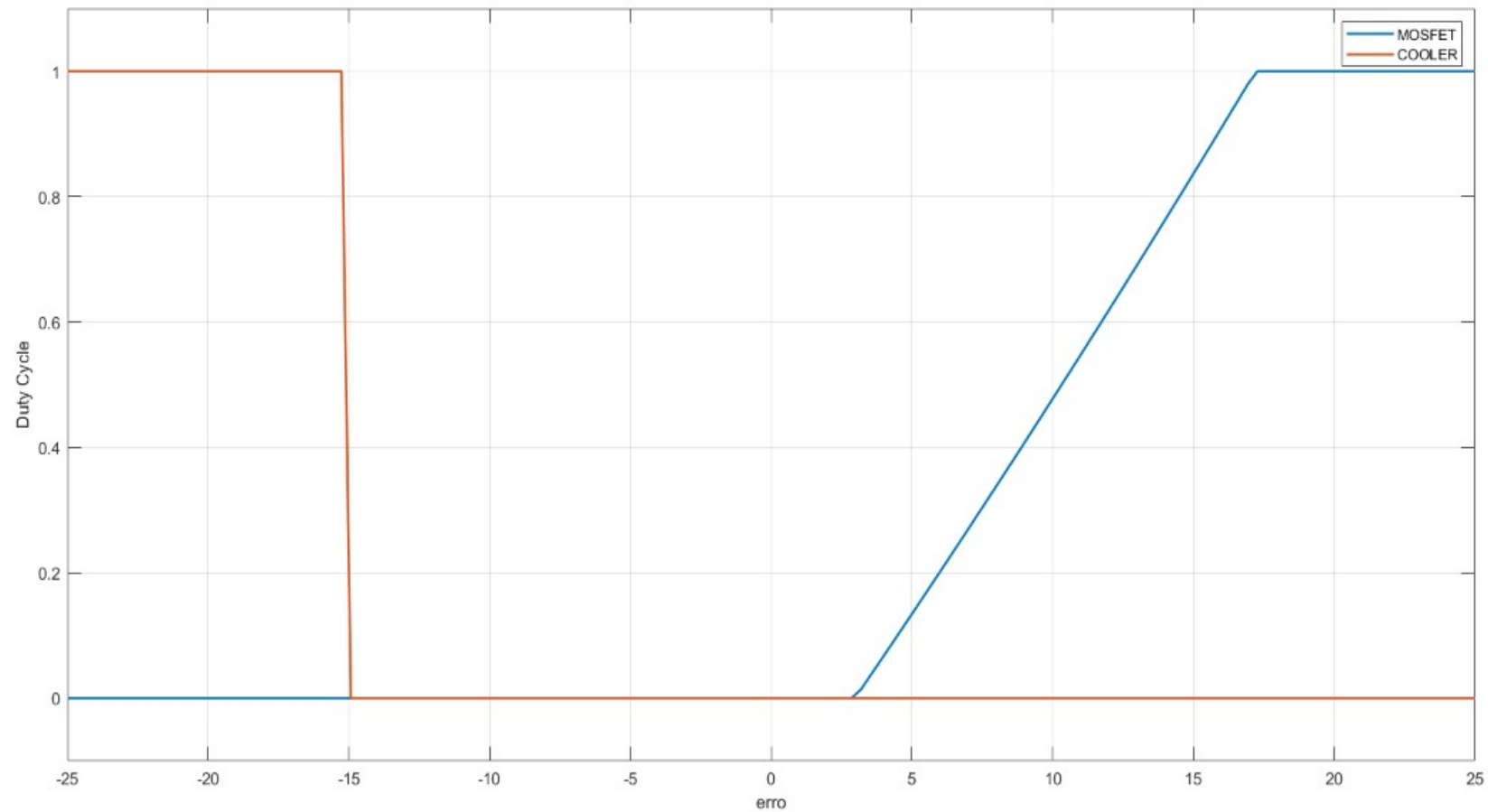
Esquemático do Sistema



Analise do Controlador

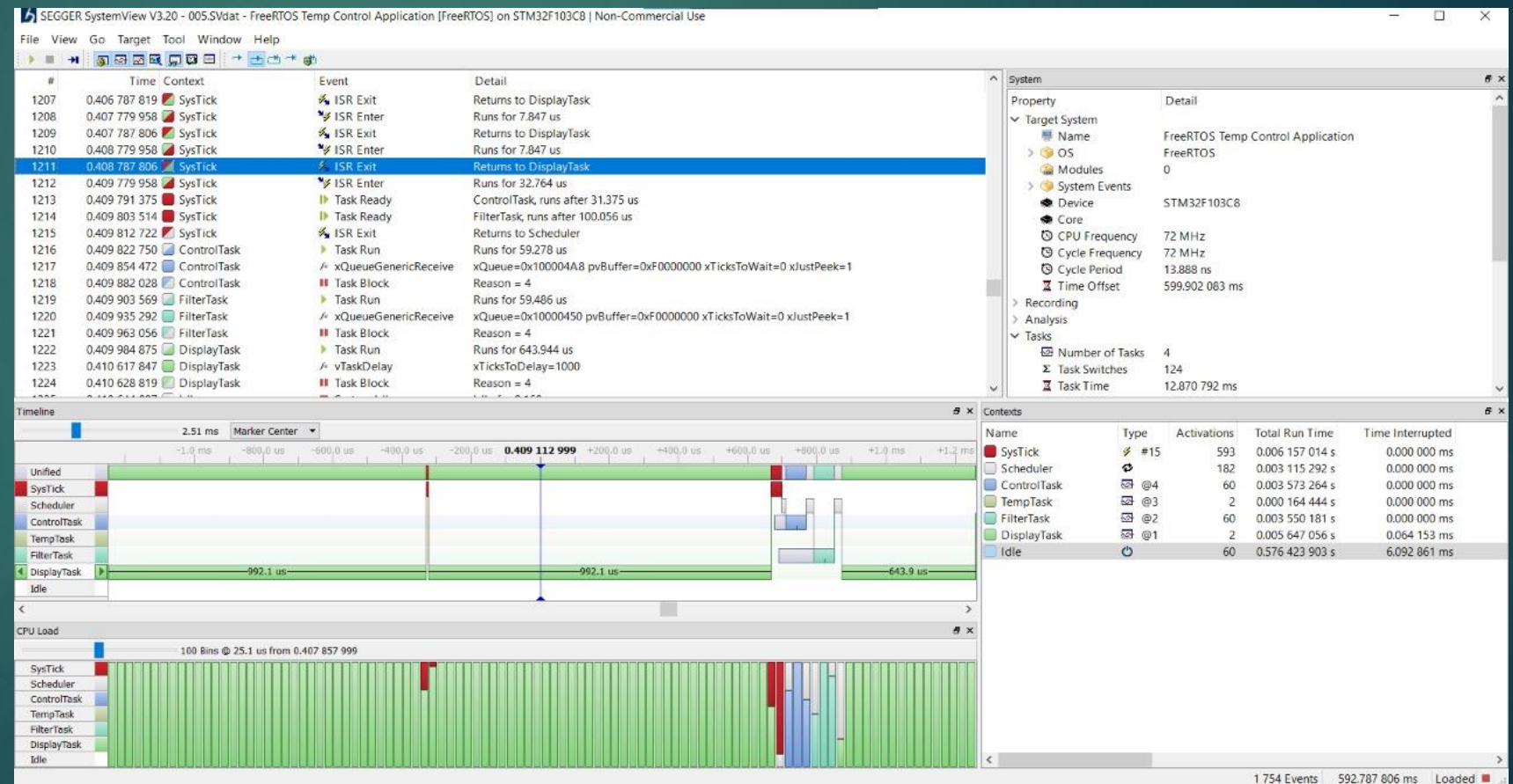


Ações de Controle

















Análise do Sistema Operacional

► Software SEGGER Systemview



Contextos do Sistema

















































Name	Type	Activations	Total Run Time	Time Interrupted
 SysTick	 #15	593	0.006 157 014 s	0.000 000 ms
 Scheduler		182	0.003 115 292 s	0.000 000 ms
 ControlTask	 @4	60	0.003 573 264 s	0.000 000 ms
 TempTask	 @3	2	0.000 164 444 s	0.000 000 ms
 FilterTask	 @2	60	0.003 550 181 s	0.000 000 ms
 DisplayTask	 @1	2	0.005 647 056 s	0.064 153 ms
 Idle		60	0.576 423 903 s	6.092 861 ms

Linha de Tempo da Aplicação

► Preempção da tarefa Display_Task

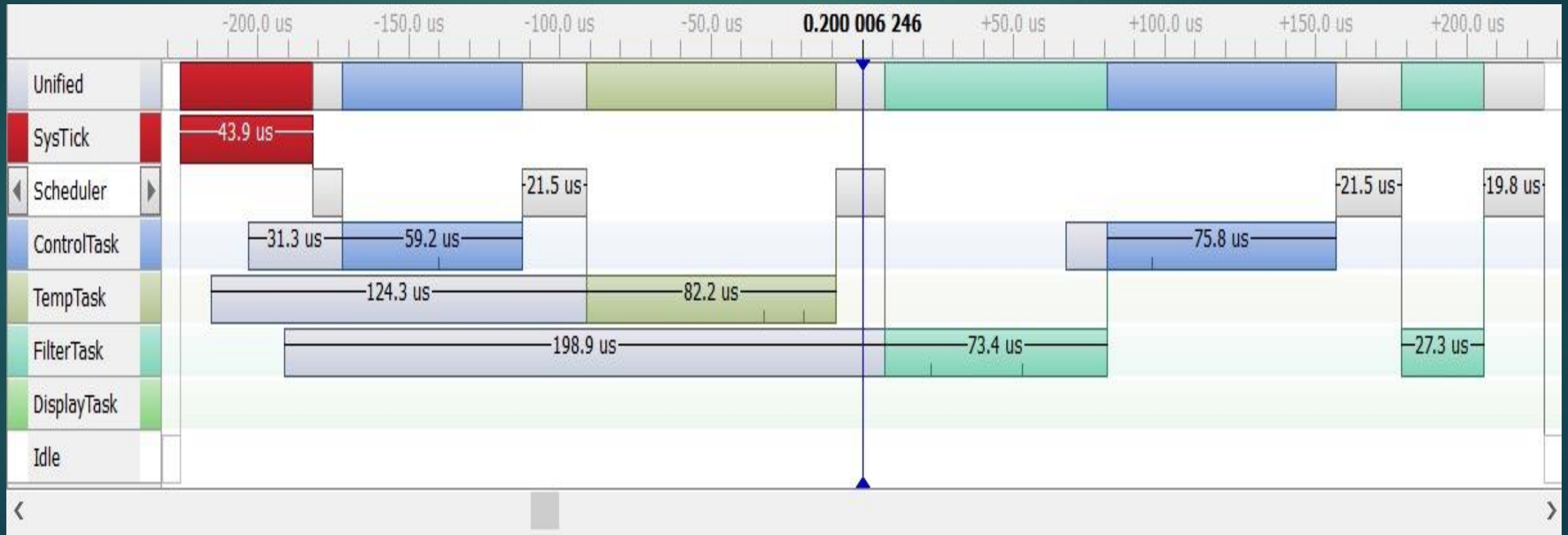


Trocas de Contexto

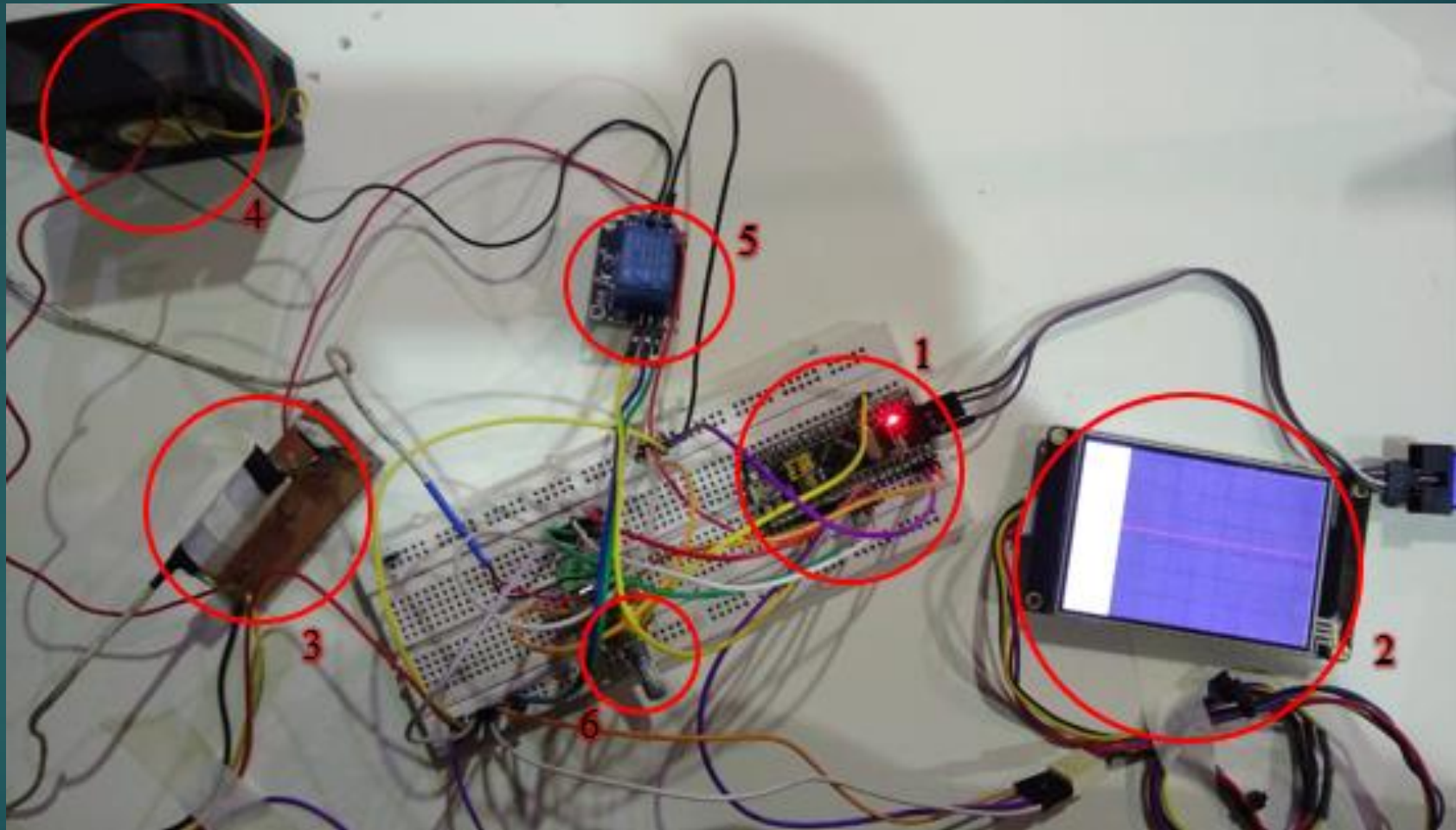
#	Time	Context	Event	Detail
1202	0.404 799 583	 SysTick	 ISR Exit	Returns to Scheduler
1203	0.404 809 611	 DisplayTask	 Task Run	Runs for 5.003 ms
1204	0.405 779 958	 SysTick	 ISR Enter	Runs for 7.847 us
1205	0.405 787 806	 SysTick	 ISR Exit	Returns to DisplayTask
1206	0.406 779 972	 SysTick	 ISR Enter	Runs for 7.847 us
1207	0.406 787 819	 SysTick	 ISR Exit	Returns to DisplayTask
1208	0.407 779 958	 SysTick	 ISR Enter	Runs for 7.847 us
1209	0.407 787 806	 SysTick	 ISR Exit	Returns to DisplayTask
1210	0.408 779 958	 SysTick	 ISR Enter	Runs for 7.847 us
1211	0.408 787 806	 SysTick	 ISR Exit	Returns to DisplayTask
1212	0.409 779 958	 SysTick	 ISR Enter	Runs for 32.764 us
1213	0.409 791 375	 SysTick	 Task Ready	ControlTask, runs after 31.375 us
1214	0.409 803 514	 SysTick	 Task Ready	FilterTask, runs after 100.056 us
1215	0.409 812 722	 SysTick	 ISR Exit	Returns to Scheduler
1216	0.409 822 750	 ControlTask	 Task Run	Runs for 59.278 us
1217	0.409 854 472	 ControlTask	 xQueueGenericReceive	xQueue=0x100004A8 pvBuffer=0xF0000000 xTicksToWait=0 xJustPeek=1
1218	0.409 882 028	 ControlTask	 Task Block	Reason = 4
1219	0.409 903 569	 FilterTask	 Task Run	Runs for 59.486 us
1220	0.409 935 292	 FilterTask	 xQueueGenericReceive	xQueue=0x10000450 pvBuffer=0xF0000000 xTicksToWait=0 xJustPeek=1
1221	0.409 963 056	 FilterTask	 Task Block	Reason = 4
1222	0.409 984 875	 DisplayTask	 Task Run	Runs for 643.944 us
1223	0.410 617 847	 DisplayTask	 vTaskDelay	xTicksToDelay=1000
1224	0.410 628 819	 DisplayTask	 Task Block	Reason = 4
1225	0.410 644 097	 Idle	 System Idle	Idle for 9.168 ms

Análise de Prioridades

- ▶ Execução conforme as prioridades



Montagem do Sistema





Obrigado pela atenção.