Critical Systems Lab - MESCC
Water Pumping Automated System

Ricardo Mendes
1201779

Arthur Gerbelli
1220201

ISEP, January 2024, **Third Delivery**

Contents

1 Introduction

This is the last report that will summarize all the analysis and implementation of the final work.

We tried to focus on the conclusions and not bring back all the specificities of the previews analyses.

Although the following chapter are tightly defined, the report should be read as an whole.

2 Requirement Specifications

- ligar uma bomba de modo aleatorio de modo a nao corruer apenas uma. - lista do que foi implementado

2.1 System Requirements

2.2 System Structure and Traceability

3 Implementation

3.1 CCSYA - Assembly

- mostrar codigo da implementacao - um dos comentarios no codigo de assembly está mal (beq)

3.2 RTAES - Concorrency and Real Time Scheduling

3.2.1 Real Time Scheduling

Table 1: Theoretical values (left) and implemented values (right) in *ms*:

Task	**Ci**	**Ti**
1	60	100
2	25	200
3	35	250

Task	**Ci**	**Ti**
1	1200	2000
2	500	4000
3	700	5000

The values on the right side table were found by running the code in different situations.

- mostrar codigo da implementacao - onde mutex

```
for (;;) {
    unsigned long start_time = millis();
    unsigned long finish_time;
    unsigned long duration;

    // Code -------------------

    finish_time = millis();
    duration = finish_time - start_time;
    Serial.print("TASK N - Execution Time[ms]: ");
    Serial.println(duration);
}
```

```
TASK 3 - Duration[ms]: 40
TASK 1 - Duration[ms]: 917
TASK 2 - Duration[ms]: 1
TASK 1 - Duration[ms]: 658
TASK 1 - Duration[ms]: 754
TASK 3 - Duration[ms]: 770
TASK 2 - Duration[ms]: 0
TASK 1 - Duration[ms]: 649
TASK 1 - Duration[ms]: 799
TASK 3 - Duration[ms]: 29
TASK 2 - Duration[ms]: 0
TASK 1 - Duration[ms]: 176
TASK 1 - Duration[ms]: 997
TASK 2 - Duration[ms]: 0
TASK 1 - Duration[ms]: 187
TASK 3 - Duration[ms]: 157
TASK 1 - Duration[ms]: 990
TASK 2 - Duration[ms]: 1
TASK 1 - Duration[ms]: 1036
```

```
long next_release = 5000 - duration;
if (next_release > 0) {
  vTaskDelay(next_release / portTICK_PERIOD_MS);
} else {
  Serial.println("TRASK 3 - FAILED Deadline !!!");
}
```

```
TASK 1 - Connection failed: 172.20.10.13
TASK 1 - Connection failed: 172.20.10.5
TASK 1 - Duration[ms]: 2009
TASK 1 - FAILED Deadline !!!
```

connection timeout = 1s that gives a lot of timeouts. specially with one of the sensors mitigation.

3.2.2 Concorrency

3.3 COMCS - Communication

- TCP pull strategy
  - Arthur

3.4 Prototype

3.4.1 Overview

- diagram if implementation

3.4.2 WPS

- explicar a tabela dos inputs dos sensores e levantar alarm caso (A tabela agora é diferente da ultima vez) - strategy for cluster of control unit nodes
  - dois timeouts dos sensores equivale a max level.... nao devia estar assim

3.4.3 MQTT Broker

- one topic per WPS

3.4.4 RSS

- reads broker every 1 sec - sends alert after 15 sec without message - subscribes to each WPS topics (iter) - log who clicked the button and maintain LED on... because the system heals it self

3.4.5 Web Server

4 Team Work

References

[1] Espressif documentation: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/system/freertos_idf.html#id23/