

INSTITUTO SUPERIOR TÉCNICO



SISTEMAS COMPUTACIONAIS EMBEBIDOS

WEATHER STATION

<i>Autores:</i>	Grupo 4	<i>Números:</i>
Diogo Moura		86976
Diogo Alves		86980
Luís Crespo		87057

10 de Dezembro de 2019

1 Estruturas de dados

Para representar o Ring Buffer, é utilizada a variável *RingBuffer*, que consiste num vetor do tipo *register_* com *NRBUF* posições, em que o tipo de dados *register_* é definido da seguinte forma:

```
typedef struct register_  
{  
    unsigned char hours;  
    unsigned char minutes;  
    unsigned char seconds;  
    unsigned char temperature;  
    unsigned char luminosity;  
} register_;
```

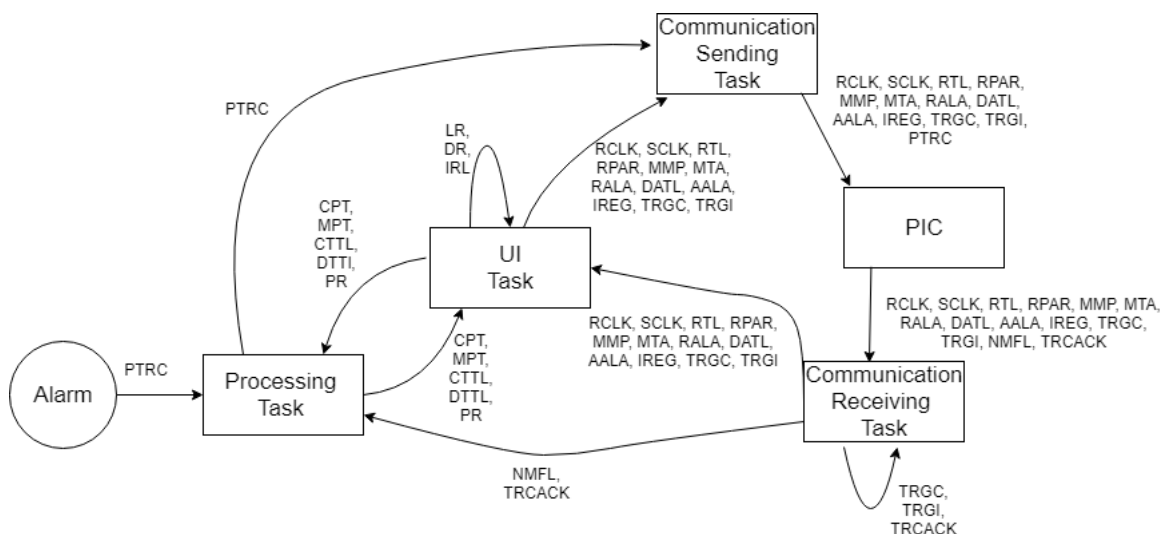
O Ring Buffer da placa é guardado na EEPROM bem como o write index (*write_index*), read index (*read_index*), número válido de registos no buffer (*nr*), próximo registo a ser transferido (*iread*) e o número de registos não transferidos (*memory*). De salientar que, para simplificar, o *read_index* é inicializado a 0 e o *write_index* é inicializado a 1 quando não são utilizados os valores guardados na EEPROM.

2 Comunicação

A comunicação entre threads é feita através de mailboxes, utilizando o mesmo protocolo que é utilizado para a comunicação com a placa:

< SOM >< CMD >< ARG1 >< ARG2 >< ... >< EOM >

em que cada parcela corresponde a um byte. Em seguida encontra-se um diagrama das mensagens trocadas:



O conteúdo das mensagens recebidas na placa é guardado num buffer que depois é processado na função **process_message**.

3 Sincronização

Os mecanismos de sincronização utilizados, para além das mailboxes, são dois mutexes - um para garantir exclusividade na escrita no ecrã e outro para garantir exclusividade no acesso ao Ring Buffer, bem como às variáveis associadas (*iwrite*, *iread*, *nr*),

4 Transferência Periódica

Para a transferência periódica de registos, é utilizado um alarme que é ativado a cada *period_of_transfer* minutos. Este alarme sinaliza a processing task para que esta emita a mensagem para iniciar a transferência periódica de registos.