

Tutte le richieste possono essere svolte a livello di Phyton

Alta priorità:

- nel received_data.txt compare ogni tanto "===== \n USART Initialized!" (due delle linee di output del comando reset), può essere spostato nel command_log.txt e non nel received_data.txt? Se sì a livello di python, come?

Nel phyton si controlla il primo carattere se è > viene buttato nei comandi se è un altro carattere viene buttato nel data. Per risolvere nel phyton si può fare una condizione del tipo: se il primo carattere è > oppure è = oppure è una TAB (perché ci potrebbe essere anche quella) allora butta nei comandi.

- gettemp ritorna due temperature (micro e sonda), differiscono di sei gradi (uController più bassa), è atteso?

E' possibile che siano diverse perché la sonda è vicino alla parte di alimentazione quindi sia maggiore.

- gettemp ritorna temperatura sonda, voltaggio, temperatura uController, come possiamo separare le info in 2 (temperature/tensione) o addirittura 3?

A livello di Phyton potete fare un parse della stringa ricevuta

- le soglie di temperatura (undert e overt) per il bit di stato a quale temperatura si riferiscono? Sonda o uController?

Sonda

- "setdac CHN valore" command_log.txt non viene scritto nel log (solo ">ACK set DAC"), come fare a scrivere nel log tutta l'informazione quando tale comando viene inviato?

Quando invio il comando scrivo una stringa nel file.

- il messaggio "Data req with Count Deactivated!" viene scritto ogni PERIODIC_TRIGGER_PERIOD, come fare a rallentarlo (ad esempio ogni qualche minuto invece che ogni PERIODIC_TRIGGER_PERIOD) per non riempire il file command_log.txt quando la scheda è attiva e in comunicazione ma senza acquisire?

Va messa una variabile globale gestibile attraverso semafori.

- quando si scrive un comando nel terminale e nello stesso momento parte uno "sparo" di getdata l'output si sovrappone e va cancellato il comando (che però rimane invisibile) altrimenti se lo si riscrive il "pezzo" di comando prima e quello dopo si sommano e tale comando non viene accettato, è risolvibile a livello di python?

Va fatta un'interfaccia grafica in phyton con la gestione dei thread

Bassa priorità:

- come funziona esattamente il threading.Semaphore?

E' il classico semaforo, se il thread A vuole scrivere su una variabile, prende il semaforo e fa quello che deve fare sulla variabile in modo che il thread B non possa agire su quella variabile fino a che il semaforo non è libero.