**Relazione progetto arduino Yùn**

Dopo aver studiato l'arduino Yun, per prima ci siamo collegati alla rete WiFi dell'arduino e lo abbiamo configurato con la rete ethernet dell'I.S.T.I.:

* Static address IPv4: 146.48.81.190
* IPv4 netmask: 255.255.248.0
* IPv4 gateway: 146.48.80.1
* DNS server: 146.48.80.3 | 146.48.80.4

Successivamente abbiamo modificato il nome e la password dello Yùn:

* Yùn name: Foda
* Password: ltfodalt

Abbiamo verificato il corretto funzionamento della configurazione provando a pingare arduino dal terminale. In seguito ci siamo connessi al WebServer di linino mediante l'url <http://146.48.81.190/sd> e abbiamo verificato la presenza di file precedentemente inseriti tramite WinSCP e puTTY.

Successivamente è stato richiesto di verificare il funzionamento di una pagina html sulla sd (raggiungibile da [http://146.48.81.190/sd HYPERLINK "http://146.48.81.190/sd/tachimetro.htm"/tachimetro.htm](http://146.48.81.190/sd/tachimetro.htm)), che avrebbe dovuto mandare a video il valore presente su un file di testo sulla sd (questo file di testo viene costantemente aggiornato da uno skatch che scrive valori casuali su di esso), quindi:

* Abbiamo creato la pagina web e i relativi file javascript(gauge.js, ld.js) per leggere dal file di testo e mandare a video sul tachimetro il valore dentro il txt.
* Abbiamo creato uno skatch che generasse, nella directory della pagina html, un file di testo con un valore intero casuale.
* Ci siamo connessi alla pagina html e tutto è andato come previsto.

A questo punto abbiamo eseguito test sulla velocità di creazione del file di testo tramite la libreria FileIO.h che utilizza la Bridge.h: la comunicazione tra arduino e linino tramite Bridge è risultata troppo lenta (300ms).

Per risolvere questo problema è stato deciso di utilizzare direttamente la porta seriale (ttyATH0) per velocizzare la comunicazione arduino-linino. Per far questo:

* Abbiamo modificato il file inittab (/root/etc/inittab) commentando la riga che indica a linux di utilizzare la porta seriale per il monitor seriale.
* Abbiamo eseguito il reboot.

A questo punto abbiamo caricato su arduino uno skatch che scrivesse sulla porta seriale una serie di valori di debug, mentre dalla parte di linino, abbiamo caricato un programma python che legge un carattere e lo stampa.

Per far girare il programma python su linino è stato necessario installare le librerie sys e serial.

Abbiamo misurato le varie velocità di trasmissione a diverse velocità, ottenendo il miglior risultato con la velocità 115200 baud, che riesce a trasmettere 1 carattere ogni 80 micro secondi.

A questo punto abbiamo iniziato a lavorare sul progetto presente sull'arduino Uno. Abbiamo studiato la logica e le strutture di dato del programma ed è stato deciso di non utilizzare multitasking sia per problemi legati alla compilazione della libreria SchedulerARMAVR.h sia perchè l'implementazione del programma per lo Yùn sarebbe stata più facile.

Per concludere abbiamo caricato il programma parzialmente modificato sull'arduino, ed abbiamo inserito un segmento di codice in grado di generare una stringa di 64 caratteri in cui abbiamo inserito il massimo ritardo sulla trasmissione arduino-linino, il valore della lettura mediante una fotoresistenza inserita sul pin analogico A3, e un padding di completamento con caratteri casuali e un ritorno carrello finale.

**10/02/2017**  **Lorenzo Dinelli, Tommaso Salvetti**