Michieletto Francesco

5B Informatica IIS Einaudi Scarpa – A.S. 2023-24

Documento di descrizione del "Capolavoro" e del percorso per realizzarlo

Progetto FM-ArtNetNode

Link al progetto: github.com/michifx512/FM-ArtNetNode_4U_V2

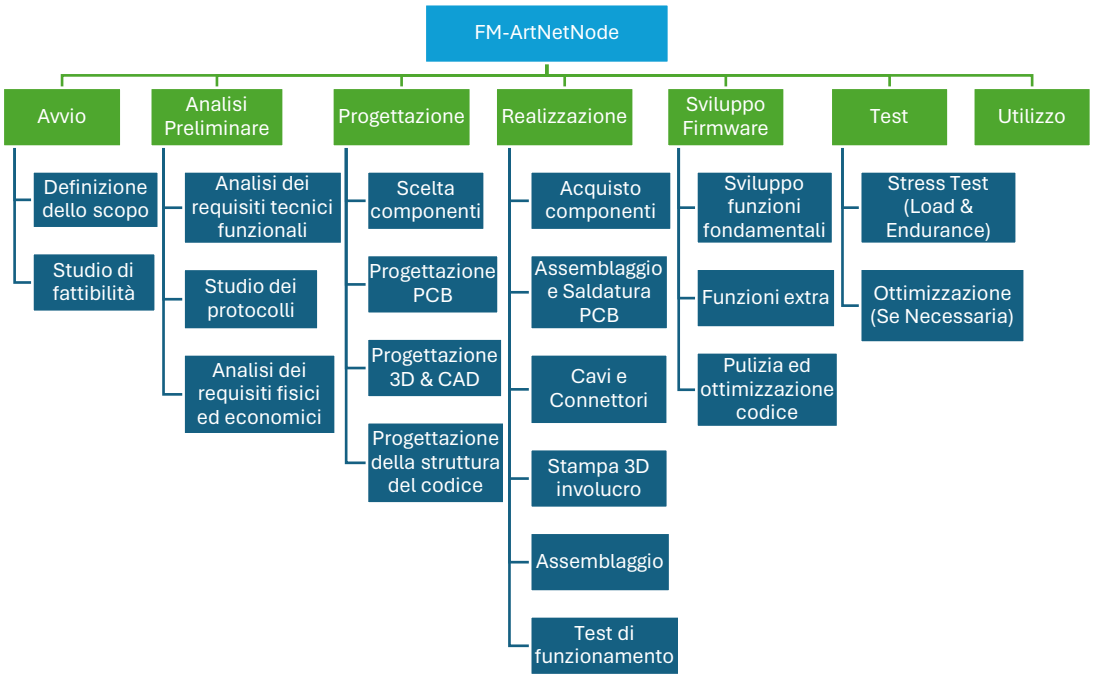
1. Introduzione

Il mio capolavoro, sviluppato in ambito extrascolastico, consiste nella completa progettazione e realizzazione di un dispositivo (chiamato tecnicamente "Nodo ArtNet") atto ad analizzare protocolli Ethernet, in specifico appunto ArtNet (protocollo UDP utilizzato per controllare sistemi di illuminazione per eventi e concerti, palchi), processarli e spedire in output su un bus DMX (protocollo riconosciuto dai corpi illuminanti stessi) i dati estratti dai pacchetti ArtNet.

Ho sviluppato le varie fasi, dall'analisi alla prima progettazione della scheda PCB (circuito stampato) con scelta e studio dei componenti necessari, le varie possibilità, pro e contro, studio di eventuali problemi e prevenzione di essi, alla sua realizzazione, programmazione, test e utilizzo.

Inoltre, è uno strumento effettivamente utilizzato nel controllo luci in eventi / concerti / discoteche.

2. Percorso di realizzazione



2.1 – Avvio e Analisi preliminare

Essendo un dispositivo con uno scopo ben definito, prima di partire con progettazione e realizzazione bisogna fare una prima analisi, secondo me uno dei punti più importanti: bisogna avere ben chiaro cosa si intenderà fare e come.
Mi sono posto queste domande, con relative risposte:

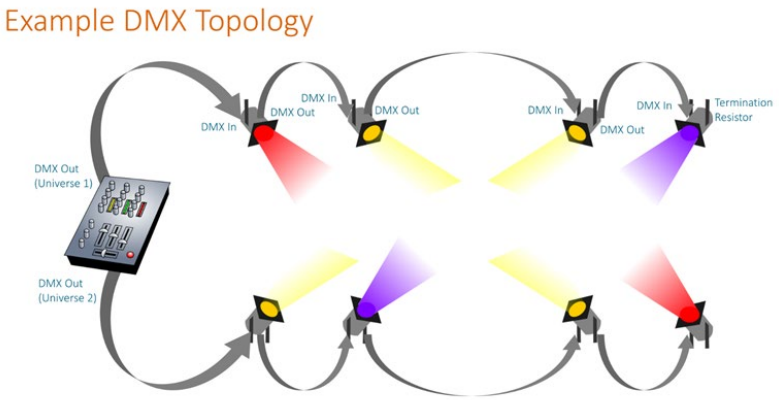
- **A cosa mi serve?**
  Mi serve a poter controllare luci, da software e consolle per il controllo, ai corpi effettivi

- **Come funziona la comunicazione i corpi illuminanti?**
  La comunicazione sfrutta il protocollo DMX, noto anche come DMX512 o DMX512-A, è uno standard industriale utilizzato per il controllo dell'illuminazione. È ampiamente utilizzato in applicazioni di illuminazione scenica e architettonica, discoteche, teatri, palchi.

  Il segnale viene inviato su un singolo cavo, con fino a 512 canali (o "slot") [essenzialmente valori di 8 bit] di informazioni. Questo è definito "universo" DMX.

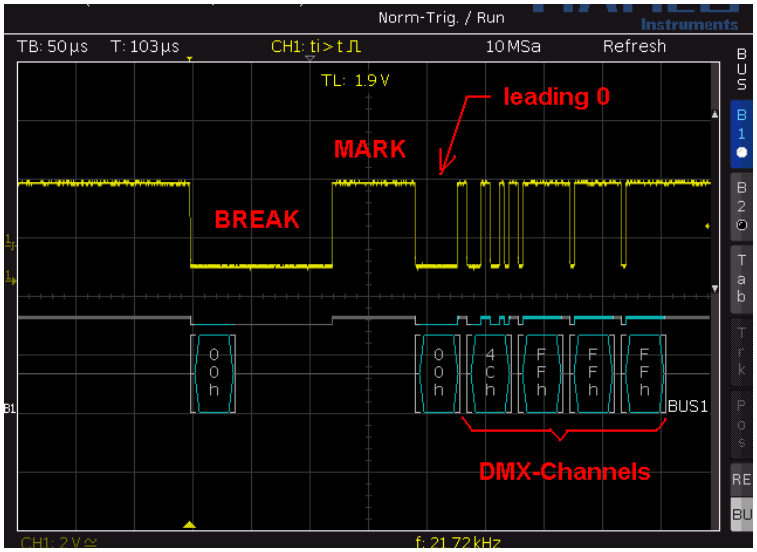
  Ogni corpo illuminante ha un ingresso DMX e un'uscita DMX, i dispositivi sono quindi collegati in cascata e il segnale viene inviato in broadcast sul bus, ogni dispositivo ha un indirizzo di partenza e "ascolterà" un numero di canali (specifico per ogni corpo).
  (Esempio: una lampada RGB che assumiamo abbia 4 parametri controllabili: Dimmer (intensità), Colore Rosso, Colore Verde, Colore Blu (ogni canale è un intero di 8 bit, quindi valori da 0 a 255)) Quindi "ascolterà" questi 4 canali, ignorando tutto il resto.

  Ogni "catena" o "cascata" di dispositivi è collegata quindi tramite un unico cavo (qui torna il concetto di universo), a una porta di un controller (in questo caso il FM-ArtNetNode).



Una semplificazione di collegamento con 2 universi

Il segnale DMX sfrutta una comunicazione differenziale EIA-485 (RS-485) come suo strato fisico. Questa informazione sarà fondamentale nella scelta dei componenti affrontata nelle fasi successive.



L'inizio di un pacchetto DMX nel bus, visto tramite un oscilloscopio.

- **<u>Quanti universi mi servono? (Quante porte di uscita)?</u>**
  Il sistema è espandibile, ma per questo progetto ho scelto come numero 4 universi, che possono rappresentare una quantità molto alta di luci, un palco di dimensioni medio-grandi o una grande sala di una discoteca.

  Considerando quindi che 4 universi (4 porte) rappresentano in somma 2048 canali, si può considerare un buon numero di corpi illuminanti controllabili.

- **<u>Che segnale ricevo in entrata e sotto quale forma?</u>**
  Il segnale che verrà ricevuto dal nodo sarà il protocollo **ArtNet v4** via cavo Ethernet.

  Art-Net è un protocollo di rete utilizzato principalmente per trasportare dati DMX512 su una rete Ethernet. Creato dalla società Artistic Licence, Art-Net si basa sullo stack di protocolli TCP/IP, utilizzando il protocollo **UDP** per il trasporto dei dati.

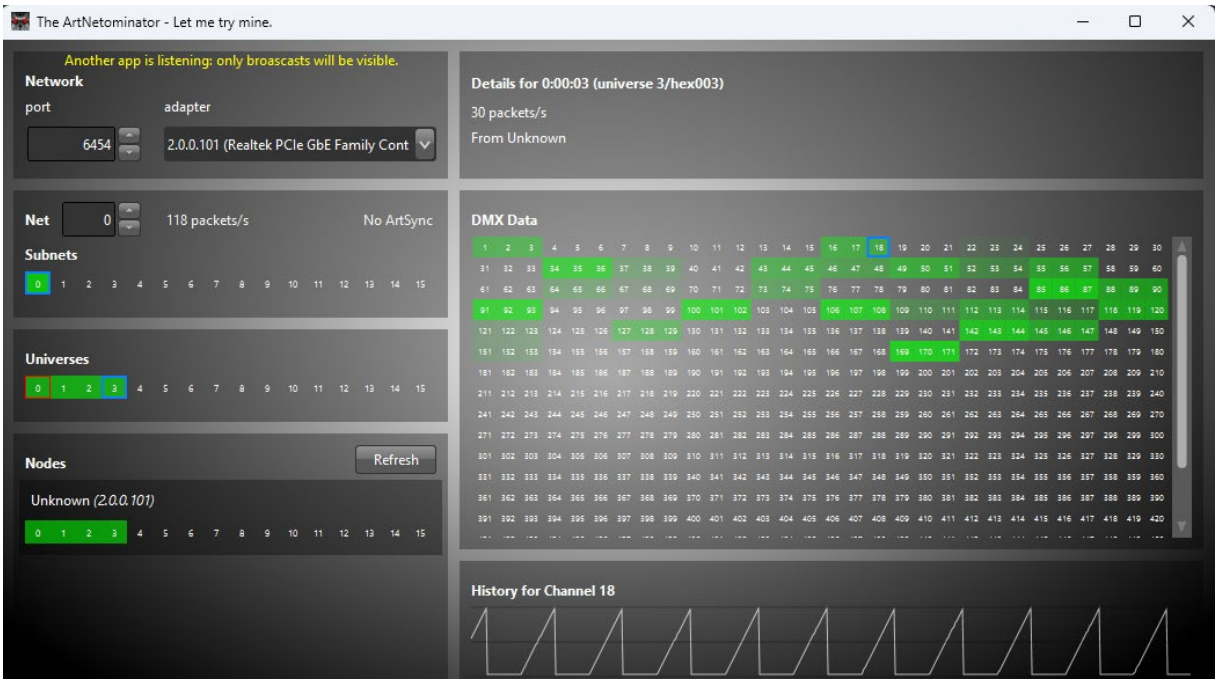
  Art-Net offre diverse potenzialità, tra cui:
  - **Scalabilità:** Può gestire un gran numero di universi DMX (fino a 32768), consentendo il controllo di un elevato numero di dispositivi di illuminazione.
  - **Flessibilità**: Essendo basato su Ethernet, Art-Net può utilizzare l'infrastruttura di rete esistente, riducendo la necessità di cablaggi specifici DMX.
  - **Velocità**: Utilizzando UDP, Art-Net è in grado di trasmettere dati in tempo reale con una latenza minima, essenziale per applicazioni di illuminazione live.
  - **Compatibilità:** Molti dispositivi e software per il controllo luci supportano Art-Net nativamente, rendendolo uno standard nell'industria dell'illuminazione.
  - **Multiplexing** o **Efficienza:** usare ArtNet semplifica le infrastrutture, perché può trasmettere anche migliaia di universi su un unico cavo ethernet (nel DMX, ogni universo è su un cavo separato) molto più comune rispetto ai cavi DMX.

Il pacchetto Art-Net DMX (ArtDmx) è composto da 530 byte totali, di cui 18 byte costituiscono l'intestazione e fino a 512 byte sono dedicati ai dati DMX (un byte per ogni canale, il valore effettivo di quel canale).

Intestazione (18 byte)

- 8 byte: ID del Protocollo ("Art-Net\0")
- 2 byte: OpCode (0x5000) (indica il tipo di messaggio, in quest'esempio indicherà i dati DMX, altri messaggi possono essere ArtPoll, ArtReply)
- 2 byte: Versione del Protocollo (solitamente 0x000E)
- 1 byte: Sequence (Utilizzato per l'ordinamento dei pacchetti DMX)
- 1 byte: Physical (Identifica la porta fisica del dispositivo DMX)
- 1 byte: SubUni (Specifica il subnet e l'universo DMX di destinazione)
- 1 byte: Net (Identifica la rete Art-Net)
- 2 byte: Length (Lunghezza dei dati DMX, massimo 0x0200 per 512 canali)

+ Dati DMX (fino a 512 byte)



Un packet sniffer ArtNet

- **<u>Limiti di dimensioni?</u>**
  I limiti sulle dimensioni complessive li ho scelti io, cercando di inserire il tutto nel minor spazio possibile. Ho scelto per comodità dimensioni di 20cm (larghezza) x 15cm (lunghezza).
  Per quanto riguarda l'altezza ho deciso di mantenere lo standard di Unità Rack, quindi 1U, ovvero circa 45mm. Questo standard è ampiamente utilizzato per tutti i dispositivi da montare appunto su armadi rack (come apparecchiature di rete, switch e router, firewall) e anche nel settore della produzione degli eventi (montaggio di splitter di segnale, mixer rack, processori di segnale, media server, nodi, switch, matrici audio e video, ...).

I limiti di lunghezza e larghezza li ho scelti basandomi anche sui limiti della dimensione della PCB (circuito stampato). Idealmente potrebbe essere di qualsiasi dimensione, ma ho scelto di distribuire nel miglior modo possibile i componenti in un'area di 100mm x 100mm.
Questo per due motivi:

1- L'azienda dove ho fatto produrre i circuiti stampati PCB (quindi JLC PCB) ha costi molto bassi per, appunto, PCB inferiori a 100mm x 100mm.
Ho scelto di far stampare la PCB da un'azienda per assicurare un'alta qualità (sono necessarie comunicazioni digitali affidabili), per semplicità, velocità, minori costi.

2- Essendo la comunicazione tra i componenti spesso su Bus SPI ad alta velocità e frequenza, con necessità di perfetta sincronizzazione, tutti i collegamenti tra un componente e l'altro devono essere il più corti possibile, e tutti della stessa lunghezza, per evitare errori di sincronizzazione, e inoltre ridurre l'attenuazione del segnale.

- **Limiti di costi?**
Gli unici limiti sui costi sono dati dal fatto che questi primi progetti non sono a scopo di commercializzazione; quindi, il budget non è molto alto. Ho potuto comunque risparmiare molto comprando i vari componenti in stock da numero abbastanza alto, e occupandomi della costruzione fisica (saldatura dei componenti sulla PCB, ...).

2.2 – Progettazione (Scelta dei componenti, disegno della PCB, disegno CAD e prima struttura del codice)

Come accennato precedentemente, uno dei componenti fondamentali sarà sicuramente l'integrato MAX485, per la comunicazione differenziale RS-485 utilizzata dal protocollo DMX.
Serviranno 4 chip MAX485, perché ogni "universo" DMX necessita una porta distinta, e quindi un chip dedicato.

Ogni MAX485 comunica con il microcontrollore via porta seriale UART; quindi, serviranno 4 porte seriali UART nel microcontrollore. Questo ha determinato la scelta di un microcontrollore come il Raspberry Pi Pico (diverso dai PI 4, questo è programmabile come Arduino, quindi in C/C++ e in Python). Ho scelto il linguaggio C/C++ per migliorare l'efficienza.

Questo microcontrollore, infatti, ha 2 porte seriali fisiche e permette di definire fino a 8 porte seriali aggiuntive usando i "Programmable I/O (PIO) state machines". Inoltre, ha ottime frequenze di clock del processore, con 133MHz e dual core.

Tra il microcontrollore, che supporta solo i 3.3V sui pin, e i MAX485 servono dei Logic Level Shifter, per permettere il cambio 3.3-5V in modo bidirezionale.

Il modulo Ethernet che ho scelto è un WizNet W5500 lite (versione lite perché ha dimensioni più piccole) che comunica con il microcontrollore via SPI. Essendo che può avere dei picchi di consumo maggiori di 200mA, ho aggiunto anche un buck converter DC-DC per poter assicurare maggior corrente, che il microcontrollore non può fornire.

Ho poi aggiunto 6 collegamenti al microcontrollore, con relative resistenze da 220Ohm, per il collegamento di piccoli diodi led, con lo scopo di segnalare l'attività e la ricezione di dati, e rendere più semplice il troubleshooting.

Infine, ho aggiunto dei condensatori in vari punti del circuito per permettere una stabilizzazione della tensione, un diodo shottky 1N5819 per assicurare una protezione dall'inversione di polarità e poter

permettere di alimentare il circuito in contemporanea con la connessione usb al microcontrollore (che entrambe a 5V potrebbero avere piccole differenze e arrecare danni).

Nella progettazione della PCB (effettuata con EasyEDA) ho inoltre aggiunto delle zone di rame per fare da schermatura e ridurre interferenze esterne.

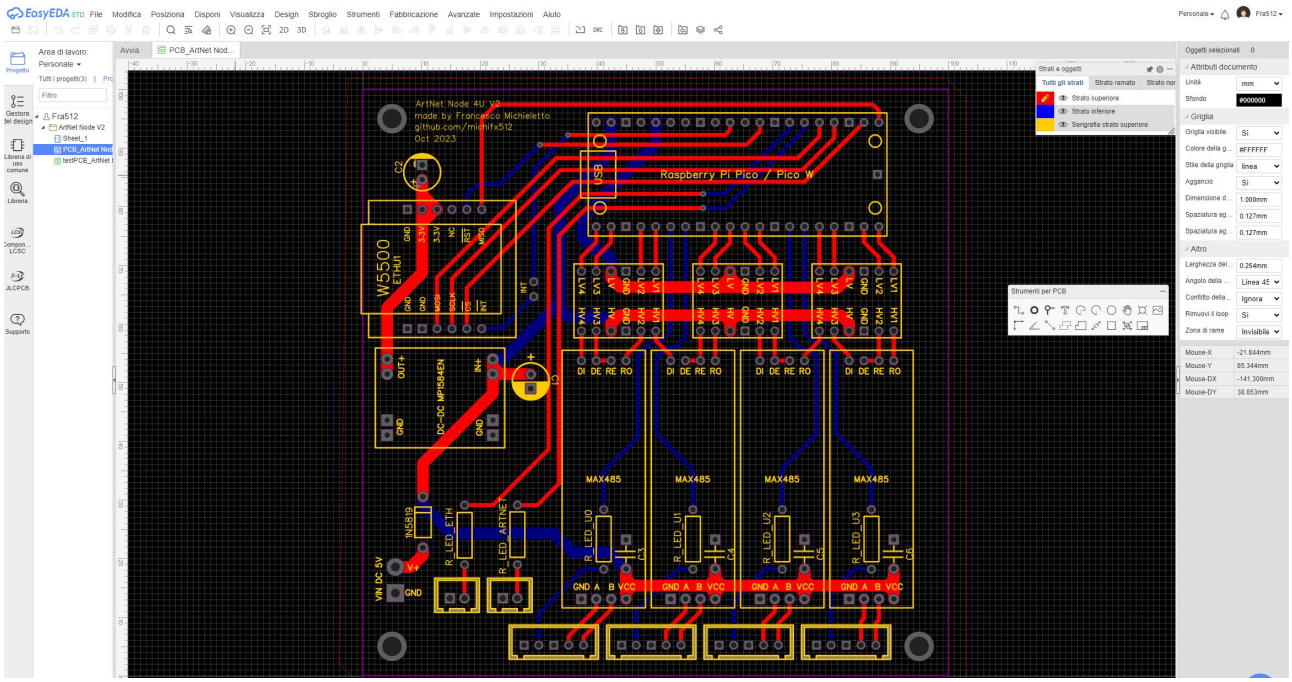
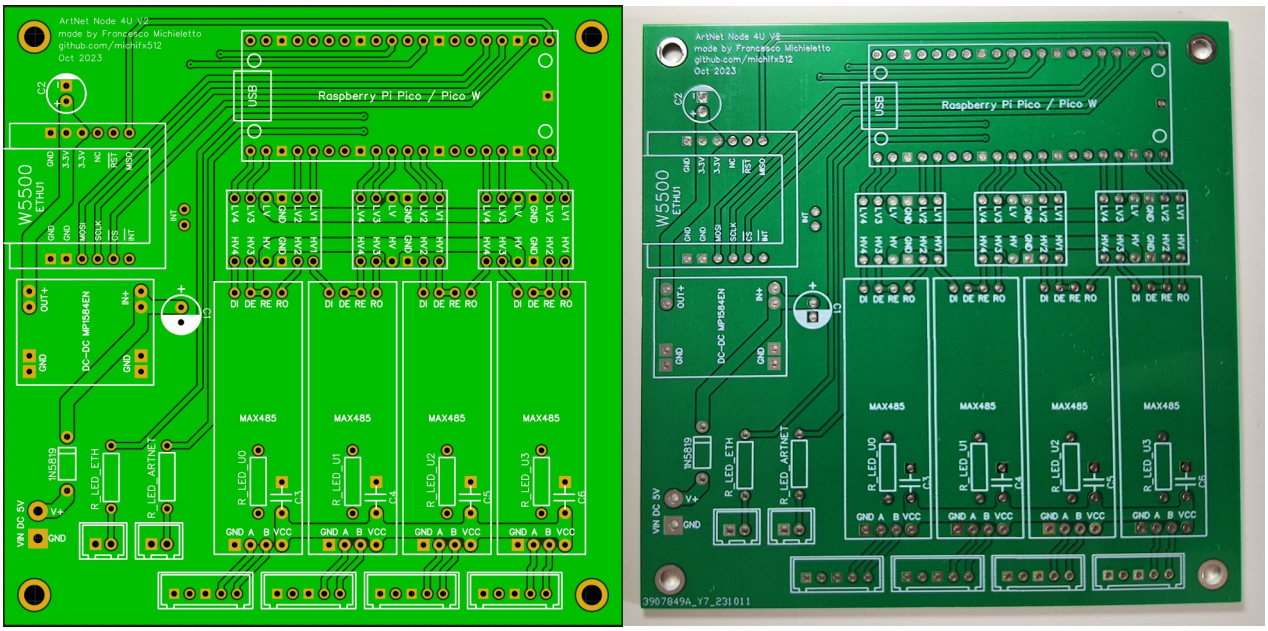
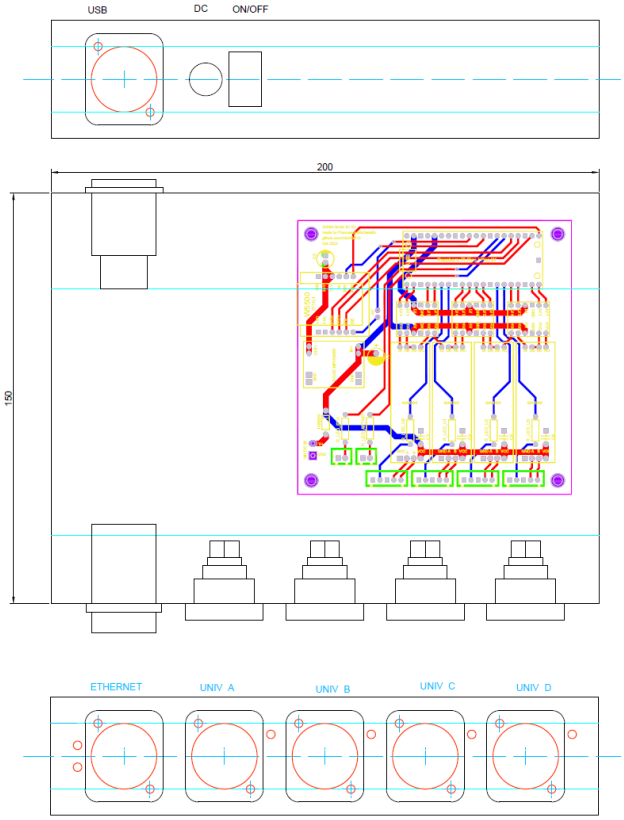


Figura 1 l'applicativo EasyEDA usato per il design della PCB, con il circuito da me creato e quella che sarà la stampa finale



A sinistra il rendering 2D della PCB prima della stampa, a destra la PCB reale



In seguito, ho realizzato un primo disegno di come sarà il dispositivo completo, decidendo posizioni dei connettori, pcb, e dimensioni effettive del nodo.

Questo servirà per la stampa 3D della scocca.

Come si può vedere in figura a sinistra, nel pannello frontale si avranno: la connessione Ethernet (con 2 led a sinistra per collegamento e ricezione dati) e le 4 porte DMX, su connettori standard XLR 3 o 5 pin.

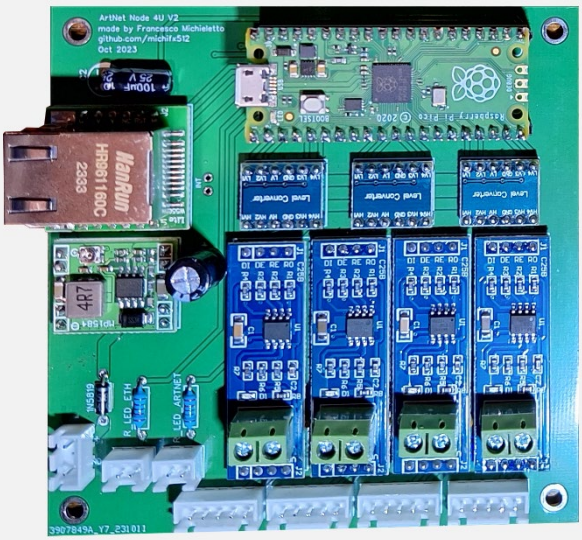
Nel pannello posteriore, invece, una presa USB tipo B femmina per potersi connettere al microcontrollore senza dover aprire il dispositivo, e permettere quindi un aggiornamento del firmware, o il troubleshooting, avendo programmato nel codice funzioni di stampa di corretto funzionamento su porta seriale usb.

Inoltre, nel pannello posteriore, un jack DC femmina per l'alimentazione, e un interruttore on/off.

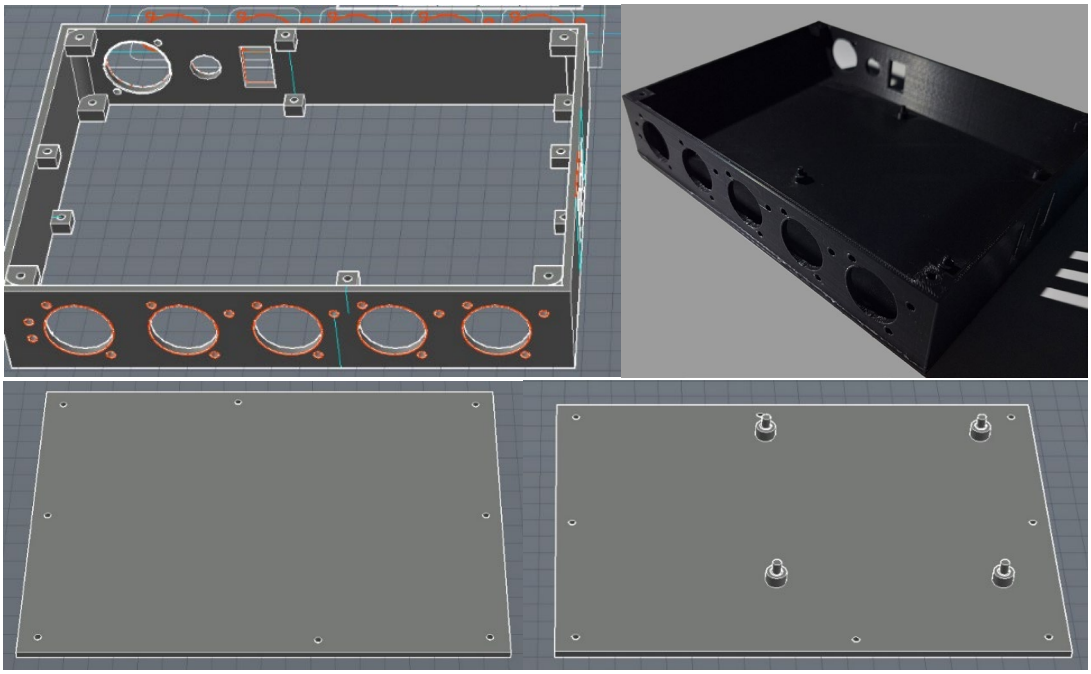
2.3 – Realizzazione Fisica, Test di Funzionamento del circuito, Programmazione del Firmware

Per quanto riguarda la realizzazione fisica ho sviluppato due parti in contemporanea:

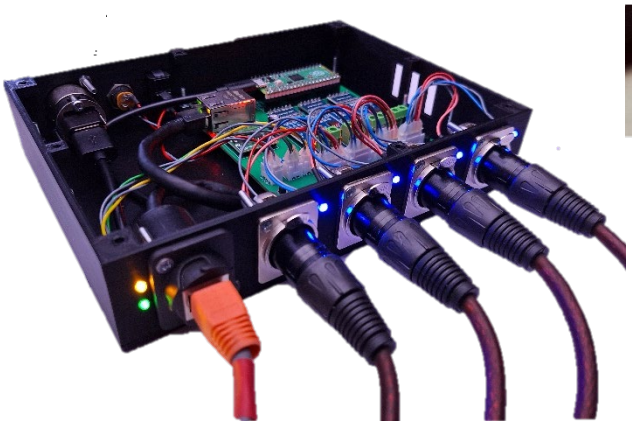
1- **Parte riguardante la PCB e l'elettronica in generale**
Ho caricato sul microcontrollore un firmware fatto abbastanza velocemente, con funzioni base giusto per fare un check dei vari componenti, ho saldato tutti i componenti sul circuito stampato, fatto un ulteriore controllo dei vari pin per sicurezza e acceso la scheda.
Verificato quindi il corretto funzionamento di essa, ho proceduto con lo sviluppo del codice effettivo e un test di esso, con gestione dei pacchetti ArtNet, output DMX, stampe di debug su seriale, led di stato.

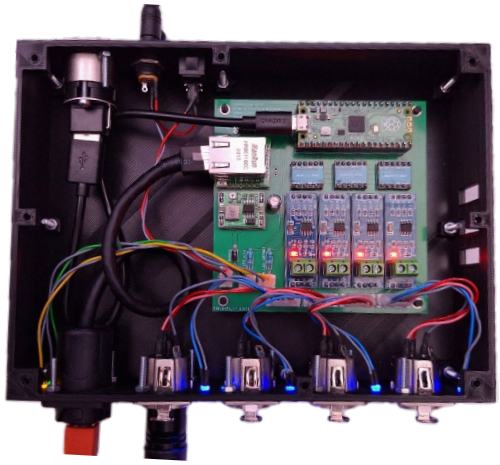


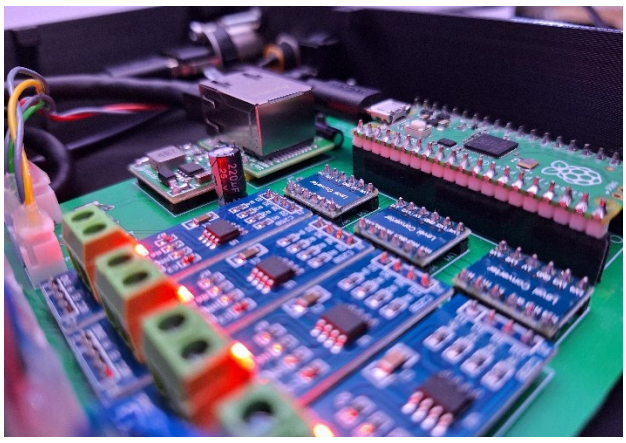
2- **Design e stampa della scocca**
Insieme alla saldatura della scheda e dei vari connettori, ho anche progettato e stampato in 3D la scocca, la scatola che contiene il tutto. Come software per il disegno ho usato AutoCAD



Il dispositivo completo è quindi il seguente:







2.4 – Test e Migliorie del Codice

L'ultima fase è dedicata al test del corretto funzionamento del dispositivo, che ho eseguito in varie modalità, con un Endurance Stress Test (di tempo, anche 4-5 giorni di continuo processing di segnale), sia con un Load Test, "bombardamento" di pacchetti, modificando il codice per far ricevere migliaia di pacchetti al secondo (in un uso massimo normale sono circa 160 frame al secondo, 40 per porta), verificando stabilità e velocità.

Verificato il corretto funzionamento resta solo la possibilità di portare aggiornamenti al codice se necessario, magari implementando funzioni extra (una funzione implementata è ad esempio la memoria dell'ultimo frame, in modo che se il nodo dovesse perdere il collegamento ethernet per qualche motivo, le luci rimarrebbero fisse nell'ultimo stato) oppure un web server integrato per la modifica di indirizzi IP, MAC (questa scheda ethernet ha la possibilità di definire un MAC), universi e porte utilizzate.

---

3. Conclusioni

Questo progetto è stato molto entusiasmante e interessante da realizzare, mettendomi in gioco per costruire uno strumento secondo le mie esigenze, studiando il funzionamento dei protocolli necessari, decidendo in pieno come costruirlo, dalla PCB al codice, alle funzioni.

Con questo progetto ho potuto sviluppare molte competenze in vari ambiti, da quelle di elettronica, (per lo studio dei vari componenti, come funzionano e di cosa hanno bisogno per funzionare), a competenze di Informatica, Sistemi e Reti (per i pacchetti ethernet), TPS (programmazione multicore), GPO con lo sviluppo in sé di un progetto, con le priorità di analisi e prevenzione dei problemi.

È stato anche coinvolgente risolvere i problemi appunto, che sarebbero potuti risultare in futuro (un esempio, in fase di progettazione ho tenuto conto da subito di possibili interferenze e reattanze nelle comunicazioni SPI, che altrimenti su un test su breadboard non avrebbero funzionato).

Ultimo, ma non per importanza, ho costruito uno strumento utile a poter controllare sistemi di illuminazione negli eventi, data la mia passione per il mondo dello spettacolo e la produzione. Dà anche soddisfazione vedere un'idea diventare realtà e sapere che un dispositivo che funziona bene lo si ha interamente progettato e costruito.

Un progetto di Francesco Michieletto