



MSc in Computer Science at University of Milan

CHIP-8 ...

Proposta per il Progetto di PROS,
corso tenuto da **Danilo Bruschi**

Email:

`federico.bruzzone@studenti.unimi.it`

`andrea.longoni3@studenti.unimi.it`

`lorenzo.ferrante1@studenti.unimi.it`

Creato da:

Federico Bruzzone

Andrea Longoni

Lorenzo Ferrante

Anno accademico 2022/2023

1 Introduzione

CHIP-8 è un linguaggio di programmazione interpretato, sviluppato da Joseph Weisbecker. Inizialmente è stato utilizzato sui microcomputer a 8 bit a metà degli anni '70. I programmi CHIP-8 vengono eseguiti su una macchina virtuale CHIP-8. È stato creato per consentire la programmazione più semplice dei videogiochi per questi computer. La semplicità di CHIP-8, la sua lunga storia e popolarità, hanno assicurato che gli emulatori e i programmi CHIP-8 siano ancora in fase di sviluppo.

Il CHIP-8 nel corso degli anni è stato esteso molte volte, tra le più importanti possiamo citare, super-CHIP-8 (SCHIP-8) e la più recente XO-CHIP.

Lo scopo del progetto è costruire un emulatore CHIP-8 in grado di eseguire tutti i giochi e i programmi scritti nel linguaggio originale e queste estensioni.

2 Componenti Hardware

Il componente principale del nostro progetto è un microcontrollore basato su architettura ARM Cortex-M4 72 MHz con 64-KB di Flash e 16-KB di SRAM, il modello esatto è STM32F334R8T6 64 pins. Utilizziamo questa scheda perché le ROM dei giochi CHIP-8 e SCHIP-8 necessitano di una dimensione massima di 4-KB di SRAM. La scheda usata durante il corso basata su architettura ARM Cortex-M0 32 MHz STM32L053 ha solo 8-KB di SRAM, quindi non sarebbe stata sufficiente per il nostro progetto, perché in SRAM dovrà girare una Virtual Machine.

Ovviamente, necessiteremo di un display TFT LCD (thin-film-transistor liquid-crystal display) a colori retroilluminato, per visualizzare il gioco. Questo display è basato sul controller ILI9341 e avente 2.4 pollici e ha una risoluzione di 320×240 px. Il display dispone di un lettore di schede microSD, che utilizzeremo per caricare i giochi dal calcolatore alla scheda.

Per interagire con il gioco, utilizzeremo una tastiera matriciale 4×4, in cui ogni tasto corrisponde ad un tasto della tastiera originale. La tastiera è adatta per l'uso sia con i controller da 3,3V che 5V e le dimensioni sono 69 mm × 77 mm × 1 mm.

Per sentire gli effetti sonori generati dal gioco, utilizzeremo un bipper che riproduce suoni a frequenza variabile. Utilizzeremo quello fornito dal kit di sviluppo.

In aggiunta, sarà possibile aggiungere uno slot per l'alimentazione via due batterie AAA.

3 Componenti Software

Le parti software che implementeremo sono:

1. La **Virtual Machine** (VM) esegue i programmi scritti in CHIP-8 e SCHIP-8. Inizialmente, il testing della VM verrà effettuato su utilizzando una libreria C tra **SDL2** o **Raylib** su un calcolatore x86 a 64 bit. Successivamente, verrà portata sul microcontrollore ARM Cortex-M.
2. Il **Driver Video** che permette l'interfacciamento tra il microcontrollore e il display TFT LCD. Lo schermo supporta interfacce parallele 6800/8080 a 8 bit e SPI a 3/4 fili, per ragioni di prestazioni noi implementeremo il driver utilizzando l'interfaccia parallela a 8 bit.
3. Il **Driver microSD** che permette l'interfacciamento tra il microcontrollore e il lettore della scheda microSD, verrà fatto utilizzando l'interfaccia SPI. Per questo driver utilizzeremo le librerie fornite da ST per la gestione del filesystem **FAT32** e facilitare lo sviluppo del trasferimento dei dati.

4. Il **Driver Keypad** che permette l'interfacciamento tra il microcontrollore e la tastiera matriciale 4×4. La gestione degli input da tastiera non verrà effettuata in polling, ma bensì tramite interrupt.
5. Il **Driver Sound** che permette l'interfacciamento tra il microcontrollore e il bipper. Il driver permette di generare suoni a frequenza variabile.

Come ultima parte software, implementeremo un menù all'avvio dell'emulatore per la selezione del gioco da eseguire. Abbiamo deciso di non usare librerie esterne ma di implementare direttamente una libreria per fare il rendering del font e scrivere sul framebuffer.

4 Analisi Tempi e Costi

Nome	Modello	Costo unitario	Unità	Costo
Schermo	ILI9341 2.4"	6.44	1	6.44
Matrix Array Keypad	AZ-Delivery 4×4	3.99	1	3.99
Microcontrollore	STM32 F334R8T6	14.98	1	14.98
Breadboard - Beeper	ELEGOO Electronic Kit	15.99	1	15.99
Totale				41.40€

Table 1: Materiali previsti per la costruzione del progetto. I costi indicati provengono da negozi online come Amazon, eBay e Aliexpress

Nome	Tempo di lavoro
Virtual Machine	3 settimane
Driver Video	1 settimana
Driver microSD	3 giorni
Driver Keypad	1 giorno
Driver Audio	5 giorni
Menù Game Selection	5 giorni
Software Optimization	2 settimane
Totale	min 2 mesi

Table 2: Tempi di realizzazione previsti per la realizzazione dei componenti software