



MSc in Computer Science at University of Milan

CHIP-8 STM32
Proposta per il Progetto di PROS,
corso tenuto da **Danilo Bruschi**

Email:
federico.bruzzone@studenti.unimi.it
lorenzo.ferrante1@studenti.unimi.it
andrea.longoni3@studenti.unimi.it

Creato da:
Federico Bruzzone
Lorenzo Ferrante
Andrea Longoni

1 Introduzione

CHIP-8 è un linguaggio di programmazione creato a metà degli anni '70 da Joseph Weisbecker per semplificare lo sviluppo di videogiochi per microcomputer a 8 bit. I programmi CHIP-8 vengono interpretati da una macchina virtuale che è stata estesa parecchie volte nel corso degli anni, tra le versioni più adottate citiamo S-CHIP e la più recente XO-CHIP.

La semplicità dell'interprete in aggiunta alla sua lunga storia e popolarità hanno fatto sì che emulatori e programmi CHIP-8 vengano realizzati ancora oggi. Nel corso degli anni molti videogiochi storici sono stati "portati" su CHIP-8 tra cui Pong, Space Invaders e Tetris.

Lo scopo del progetto è quello di costruire un emulatore CHIP-8 e S-CHIP in grado di funzionare su un microcontrollore STM32.

2 Analisi di mercato

Ad oggi esiste solo una specifica per la virtual machine di chip8 ed esistono numerose implementazioni di quest'ultima per quasi tutte le piattaforme, nonché alcuni strumenti di sviluppo. I giochi sono ancora sviluppati per CHIP-8 oggi, oltre ai vecchi giochi che riaffiorano online negli archivi digitali. Sebbene CHIP-8 e SCHIP siano stati comunemente implementati come emulatori, esiste anche un'implementazione hardware pura (scritta nel linguaggio Verilog) per alcune schede FPGA. Attualmente è difficile ottenere la cifra esatta di utenti di chip8, il topic "chip8" su github ha 750+ repository in più di 10 linguaggi diversi tra questi il più popolare con 600 stelle è Octo, un ide e vm (base, super, xo) online scritta in js, scritta dal inventore dell'estensione OXchip la quale nel 2014 ha riportato in vita chip8 ammodernandolo un po' e aggiungendo nuove features un'altra cosa che si può citare è la octojam, che quest'anno sarà alla decima edizione, una game jam basata su chip8 con più di 25 entry ad anno le piattaforme che lo implementano sono illimitate, siccome è una specifica con molte poche richieste hardware e con un instructionset molto piccolo uno lo ha implementato per girare su gameboy color, calcolatrici hp48, ti48, emacs

3 Componenti Hardware

Il componente principale è il microcontrollore **STM32F334R8T6** basato su architettura ARM con processore Cortex-M4 da 72 MHz, 64 Kb di memoria flash e 16 Kb di SRAM. Abbiamo deciso di utilizzare questa scheda perché le ROM dei giochi CHIP-8 e S-CHIP hanno dimensione massima di 4 Kb. Considerando questo e il fatto che la macchina virtuale necessita 4 Kb per poter funzionare non è stato potuto utilizzare la scheda fornitaci durante il corso a causa della sua quantità limitata di SRAM.

Necessiteremo inoltre di un display TFT LCD (thin-film-transistor liquid-crystal display) a colori retroilluminato, per poterci interfacciare col gioco. Questo display, da 2.4 pollici, è basato sul controller **ILI9341** e ha una risoluzione di 320×240 px.

Il display dispone di un lettore di schede microSD, che verrà utilizzato per caricare una delle ROM presenti sulla SD in SRAM.

Per interagire con l'emulatore utilizzeremo una tastiera matriciale 4×4 corrispondente alla tastiera esadecimale originale.

Per riprodurre gli effetti sonori generati dal gioco utilizzeremo un beeper a frequenza variabile. In CHIP-8 e S-CHIP vi è un solo tipo di suono che può essere riprodotto durante tutta l'esecuzione del gioco.

Infine sarà possibile aggiungere uno slot per l'alimentazione tramite due batterie AA.

4 Componenti Software

Le parti software che saranno implementate sono:

1. La **Macchina Virtuale** (VM) che interpreta programmi CHIP-8 e S-CHIP. Non volendoci affidare ad implementazioni già esistenti abbiamo deciso di svilupparla da zero, per questo motivo verrà effettuato il testing con un'interfaccia emulata su un calcolatore x86 a 64 bit tramite SDL2 o Raylib (due librerie per creare interfacce multimediali in C). Successivamente verrà portata sul microcontrollore basato su ARM Cortex-M.
2. Il **Driver Video** che permette l'interfacciamento tra il microcontrollore e il display TFT LCD. Lo schermo supporta interfacce parallele a 8 bit e SPI a 4 pin, 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 SD, verrà sviluppato 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 dati.
4. Il **Driver Keypad** che permette l'interfacciamento tra il microcontrollore e la tastiera matriciale 4x4. 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 beeper. Il driver permette di generare suoni a frequenza variabile.

Inoltre verrà implementato un menù per la selezione del gioco all'avvio dell'emulatore. Infine abbiamo deciso di non usare librerie esterne ma di sviluppare una nostra libreria per effettuare il rendering del font e disegnare sul display.

5 Architettura Software

6 Analisi del Consumo Energetico

Come abbiamo già accennato in precedenza, il microcontrollore lo alimenteremo durante lo sviluppo via USB e successivamente utilizzando batterie AA. Le batterie AA hanno una tensione di 3V e una capacità di ~2500 mA, usandone due in serie otteniamo 5000 mA. Stimando un consumo dello schermo di 100 mAh e un consumo medio di 25 mAh per il microcontrollore e le periferiche, per un totale di 125 mAh, sapendo che:

$$\begin{aligned}\text{time (h)} &= \frac{\text{capacity (mA)}}{\text{consumption (mAh)}} \\ &= \frac{2 * 2500 \text{ mA}}{125 \text{ mAh}} \\ &= 40 \text{ h}\end{aligned}$$

Possiamo stimare una durata di 40 ore di gioco.

7 Analisi di Tempi e Costi

Nome	Modello	Costo unitario	Unità	Costo
Schermo	ILI9341 2.4"	6.44	1	6.50
Matrix keypad	AZ-Delivery 4×4	3.99	1	3.99
Microcontrollore	STM32 F334R8T6	14.99	1	14.99
Beeper		0.99	1	0.99
Breadboard e cablaggio		4.99	1	4.99
Totale				31.50€

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

Nome	Tempo di lavoro
Macchina virtuale	3 settimane
Driver video	1 settimana
Driver microSD	3 giorni
Driver keypad	1 giorno
Driver audio	5 giorni
Selezione gioco	5 giorni
Ottimizzazione software	2 settimane
Totale	2 mesi

Table 2: Tempi previsti per la realizzazione dei componenti software

8 UML test

