

Arduino Gameboy

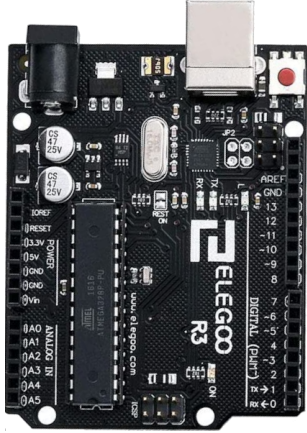

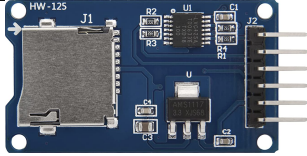

Samuele Cerea e Andrea Facoetti

May 22, 2022

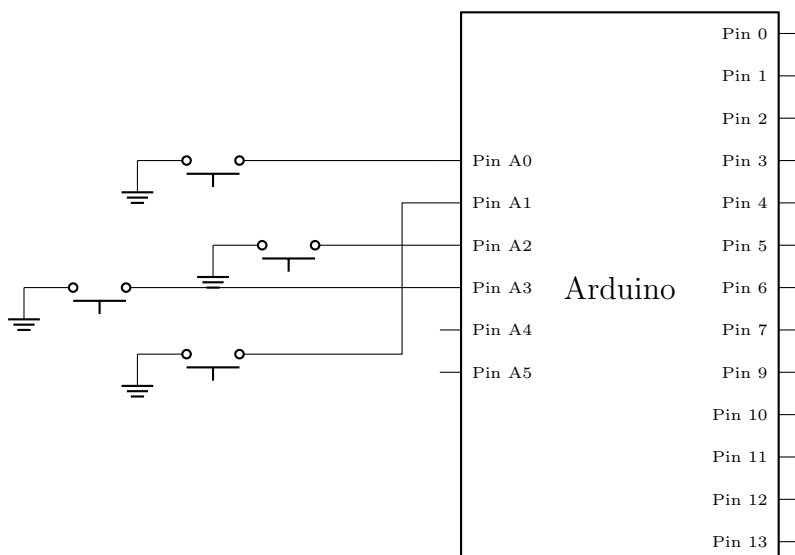
1 Obiettivo

L'obiettivo di questo progetto è quello di creare una piccola console portatile, simile al Gameboy, con un Arduino. Quindi questa console deve essere in grado di caricare dei giochi da una cartuccia e poi eseguirli.

2 Componenti

Quantità	Nome	Prezzo	Immagine
1	Arduino Uno	15,99€	
1	Kit Pulsanti	8,99€	
1	Lettore Scheda SD SPI	5,99€	
1	Display OLED 128x64 2,42 pollici SPI	31,99€	
	Cavi jumper	5,00€	
	Filo solido 22 AWG	10,00€	
3	Breadboard	5,00€	

3 Circuito



3.1 Leggere input di un bottone

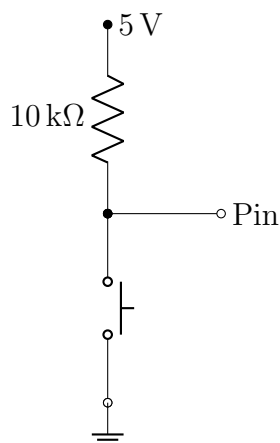


Figure 1: Pulsante in configurazione pull-up

Per leggere lo stato di un bottone è necessaria una configurazione pull-up o pull-down per evitare situazioni in cui lo stato del pin di input è indefinito.

Sulla sinistra è disegnato un pulsante in configurazione pull-up, e si può osservare che quando il pulsante è aperto il pin sarà connesso ai 5 volt attraverso la resistenza, questo assicura che il pin ha uno stato HIGH quando il pulsante non è premuto. Quando si preme il pulsante il pin viene connesso anche al ground impostando lo stato logico di LOW, inoltre la resistenza limita la corrente che passa tra il 5 volt e il ground. Quindi la resistenza deve avere un valore sufficientemente grande in modo da non sprecare corrente eccessiva, ma deve comunque far passare abbastanza corrente per permettere all'Arduino di rilevare lo stato logico HIGH quando il pulsante è aperto.

4 Codice

Il codice si trova su github [samu698/Arduino](#) Questo progetto ha due giochi snake ed una replica del gioco del dinosauro di chrome, il codice di questi due si trova in due file diversi snake.ino e dino.ino.

Questi file sono suddivisi in due parti una uguale ed una specifica al gioco, la prima parte del codice svolge varie funzioni necessarie per entrambi i giochi:

- Definizione di costanti come il numero dei pin e la dimensione dello schermo
- Inizializzazione dello schermo e dei pin
- Lettura dello stato dei bottoni
- definire la frequenza di aggiornamento dello schermo

La parte fissa del codice definisce due funzioni `setup()` e `loop()` dove la prima verrà chiamata quando l'Arduino si attiva e la seconda verrà chiamata ripetutamente dopo che viene eseguito la funzione setup.

Queste due funzioni richiedono la presenza due funzioni `gameSetup()` e `gameLoop()` in cui la prima svolgerà l'Inizializzazione del gioco e la seconda viene chiamata ogni frame per aggiornare lo stato del gioco e disegnare sullo schermo.

4.1 Leggere lo stato dei pulsanti

Lo stato dei pulsanti viene letto ripetutamente mentre il codice aspetta tra un frame ed il successivo salvando lo stato dei pulsanti in due variabili `btnsPressed` `btnsHeld` dove la prima indica quali pulsanti sono stati premuti quel frame e la seconda quali pulsanti sono tenuti premuti.

Per ridurre l'uso della memoria lo stato dei pulsanti è salvato nei bit di queste variabili. Grazie alle macro `setBit(v, x)`, `resetBit(v, x)` e `getBit(v, x)` è possibile scrivere, cancellare e leggere il bit in posizione x della variabile v. Le costanti `UP_BTN`, `DOWN_BTN`, `RIGHT_BTN` e `LEFT_BTN` indicano a quale bit corrisponde quale pulsante. Quindi per leggere se il pulsante UP è stato premuto si può usare il codice:

```
getBit(btnsPressed, UP_BTN)
```

4.2 Il bootloader

Il bootloader è del codice che si trova nella sezione finale della memoria dell'Arduino e solitamente permette di caricare i programmi quando viene

connesso al computer. È possibile modificare questo codice cambiare il funzionamento dell'Arduino.

Per questo progetto i giochi devono essere caricati da una scheda SD quindi il bootloader, al posto di caricare il codice dal computer attraverso una comunicazione seriale, li dovrà caricare da una scheda SD per questo è possibile utilizzare il bootloader avr-boot di zevero su github.

4.2.1 Come programmare il bootloader

Caricare il bootloader sull'Arduino richiede l'utilizzo di un'altro dispositivo, il programmatore, in quanto il bootloader non può riscrivere se stesso.

Il programmatore da noi utilizzato è un'altro Arduino su cui viene caricato il codice che si può trovare nella sezione esempi dell'Arduino IDE sotto la sezione Arduino ISP.

Le due schede devono essere connesse come nel diagramma seguente, e poi sempre dalla IDE Arduino è possibile caricare il bootloader.

4.3 Caricare il codice sulla scheda SD

Il bootloader quando l'Arduino viene resettato premendo il pulsante controlla che sulla scheda SD ci sia un file chiamato "FIRMWARE.BIN" (case-sensitive) e carica il codice presente al suo interno.

Il questo file si può ottenere utilizzando l'Arduino IDE per produrre un file .hex dal codice del gioco e poi utilizzare il comando seguente per convertire il file nel formato corretto

```
avr-objcopy -I ihex -O binary <INPUT>.hex FIRMWARE.BIN
```

5 Risorse e librerie utilizzate