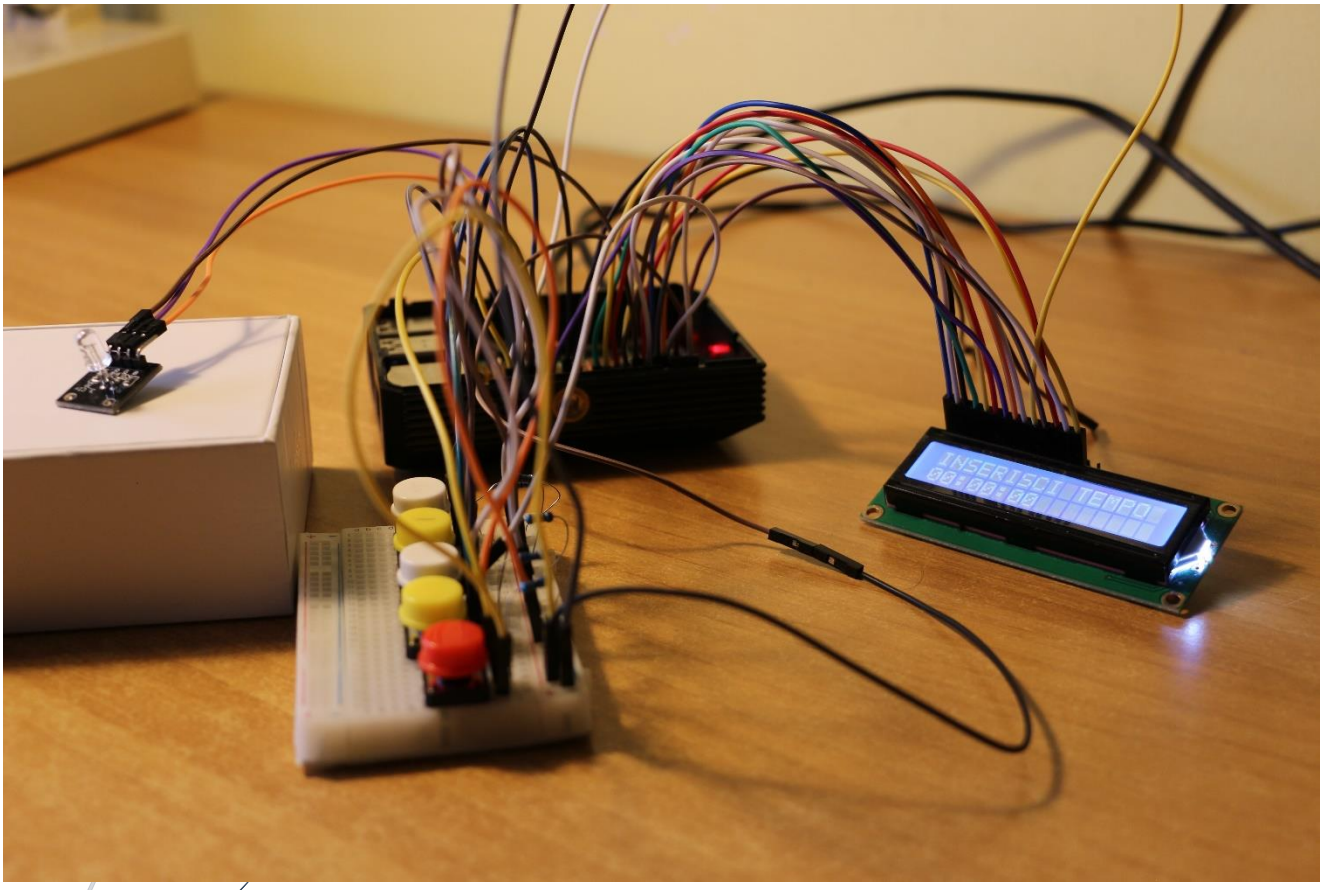
Tesina Sistemi Embedded
Progetto: CountBerry



MATERIA: SISTEMI EMBEDDED
PROFESSORE: DANIELE PERI
AA: 2020-2021
STUDENTI: GABRIELE GAMBINO , MARCO ZACCARIA

Sommario

DESCRIZIONE DEL TEAM

Il team è composto da Gabriele Gambino e Marco Zaccaria, due studenti del corso di laurea magistrale di ingegneria informatica presso l'Università degli Studi di Palermo.

CAPITOLO1 – INTRODUZIONE

Questo progetto ha l'obiettivo di realizzare un sistema, chiamato CountBerry, finalizzato ad implementare un countdown. A partire da una certa di quantità di tempo stabilita dall'utilizzatore, CountBerry eseguirà il conto alla rovescia fino allo scadere del tempo.

Lo strumento target utilizzato per realizzare il countdown è un Raspberry Pi 4B , per l'interazione con il countdown sono stati adoperati 5 pulsanti colorati, mentre per visualizzare in tempo reale lo stato del sistema sono stati adoperati un display LCD e uno schermo collegato tramite HDMI.

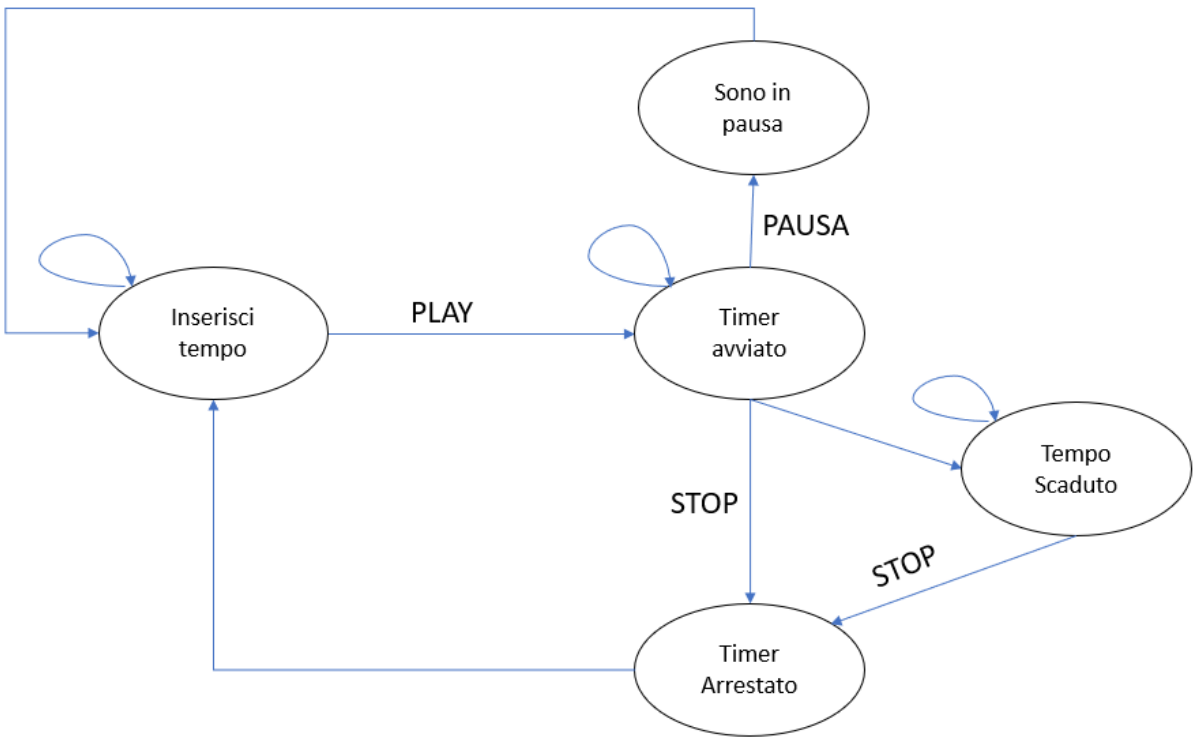
Panoramica generale

All'avvio il sistema si trova nello stato di "Inserisci Tempo" in cui attende per un tempo indefinito che l'utente, tramite i pulsanti di inserimento, inserisca il tempo da cui far partire il countdown. Dopo aver scelto il tempo, l'utente per avviare il countdown dovrà cliccare il pulsante di avvio/play. Successivamente CountBerry si trova nello stato di "Timer Avviato" in cui il tempo comincia a scorrere. Qualora l'utente dovesse cliccare sul tasto pausa , il countdown entra nello stato "Sono in Pausa" , in questo caso si fermerà, e contestualmente ritornerà allo stato iniziale di "Inserisci Tempo". L'utente così potrà far continuare il countdown dal punto in cui è stato fermato oppure aggiungere/diminuire i secondi a disposizione.

Se l'utente clicca sul tasto Stop il countdown si troverà nello stato di "Timer Arrestato" , e successivamente ritornerà allo stato iniziale.

Mentre se il tempo scade naturalmente, ovvero senza che l'utente lo fermi di proposito, lo stesso si troverà nello stato di "Tempo Scaduto" e vi rimarrà fin quando l'utente non cliccherà il tasto di Stop, a quel punto il countdown si troverà nello stato di "Timer Arrestato" , e successivamente ritornerà allo stato iniziale.

Segue l'automa degli stati finiti.



Requisiti funzionali e non funzionali

Il team nella realizzazione del progetto si è posta fin da subito quelli che erano i requisiti minimi che lo stesso dovesse presentare.

Tra i requisiti funzionali sono stati inseriti:

- Sistema target adoperato ovvero Raspberry Pi 4B;
- Visualizzazione a schermo LCD. Si adopera per tale funzione uno schermo LCD 1602;
- Pulsanti per il funzionamento. I tre pulsanti minimi scelti per la realizzazione del countdown sono avvio, pausa e stop;
- Led RGB. Adoperato per dare un avviso visivo all'utente quando il countdown sta per scadere.

Mentre tra i requisiti non funzionali:

- Breadboard: A causa dell'uso di tanti pulsanti, la mini-breadboard è stata molto d'aiuto per sistemare meglio i jumper e i pulsanti che permettono all'utente di interagire con il sistema;
- Autonomia. Il sistema proposto non ha bisogno di un computer per l'input da tastiera, una volta caricato il codice non ci sarà più bisogno di una UART per interagire con il sistema target,;
- Input del tempo: Nella prima versione del sistema, il tempo veniva inserito da tastiera, quindi attraverso l'ausilio di un computer , connesso tramite UART al sistema target. Il team volendo rendere il tutto più interattivo ha voluto inserire altri due pulsanti per l'inserimento del tempo;
- Schermo HDMI. Permette di visualizzare lo stato in cui si trova il countdown in tempo reale, attraverso una rappresentazione grafica con tre quadrati.

CAPITOLO2 – HARDWARE ADOPERATO

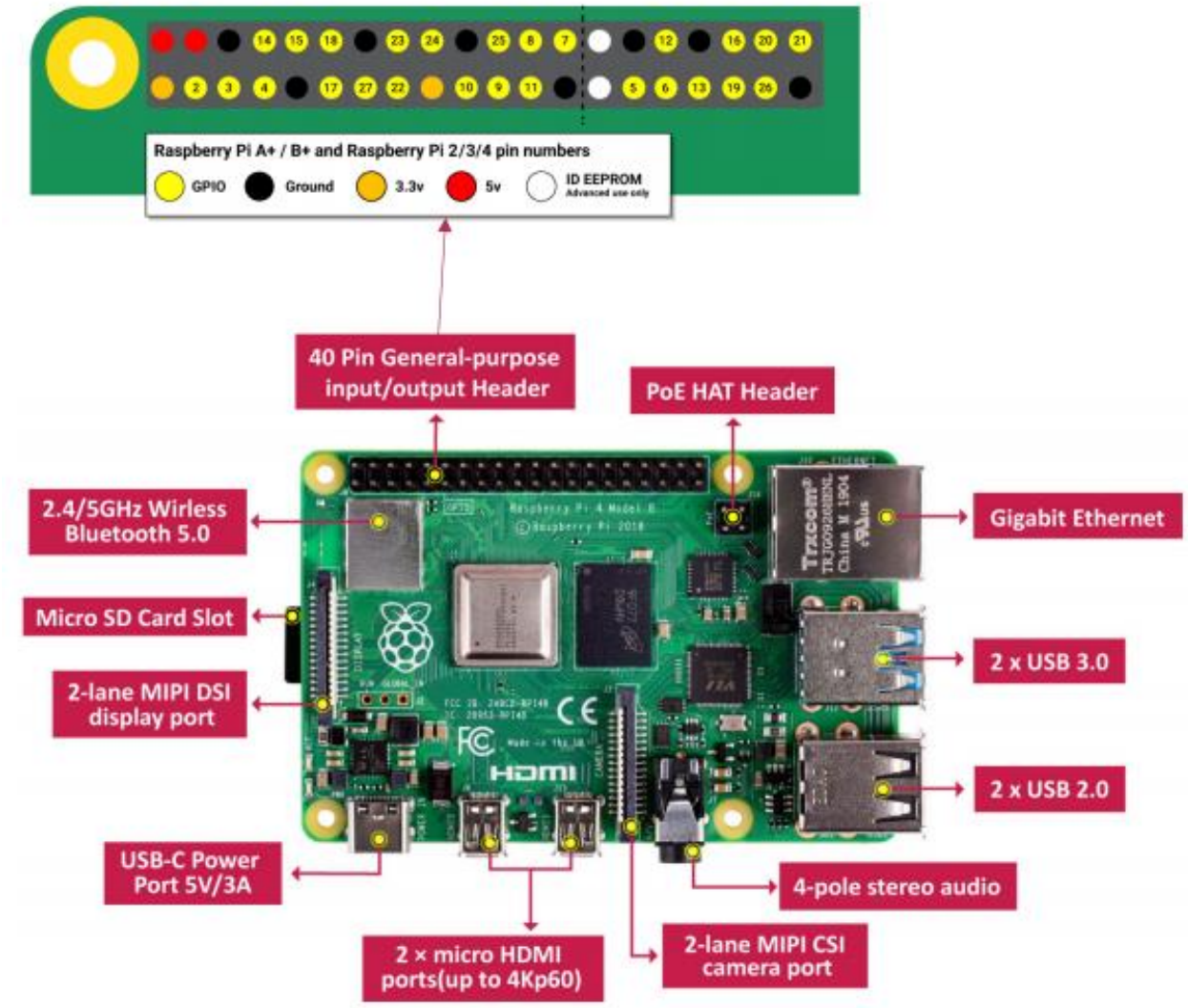
Per la realizzazione di CountBerry è stato quindi adoperato il seguente Hardware:

- Raspberry Pi 4B
- Mini-Breadboard
- Pulsanti colorati
- Resistenze e Jumper
- Schermo LCD 1602
- Schermo HDMI
- LED RGB

RASPBERRY PI 4B

Raspberry Pi 4 Model B è l'ultimo prodotto della popolare gamma di computer Raspberry Pi. Offre aumenti rivoluzionari in termini di velocità del processore, prestazioni multimediali, memoria e connettività rispetto alla generazione precedente Raspberry Pi 3 Model B pur mantenendo la compatibilità con le versioni precedenti e consumo di energia simile.

Per lo sviluppo del progetto sono state adoperate le GPIO, le alimentazioni a 5V e 3,3V e la porta micro-HDMI.



Schema Pin Raspberry

Pin#	NAME		NAME	Pin#
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1, I²C)		DC Power 5v	04
05	GPIO03 (SCL1, I²C)		Ground	06
07	GPIO04 (GPCLK0)		(TXD0, UART) GPIO14	08
09	Ground		(RXD0, UART) GPIO15	10
11	GPIO17		(PWM0) GPIO18	12
13	GPIO27		Ground	14
15	GPIO22		GPIO23	16
17	3.3v DC Power		GPIO24	18
19	GPIO10 (SPI0_MOSI)		Ground	20
21	GPIO09 (SPI0_MISO)		GPIO25	22
23	GPIO11 (SPI0_CLK)		(SPI0_CE0_N) GPIO08	24
25	Ground		(SPI0_CE1_N) GPIO07	26
27	GPIO00 (SDA0, I²C)		(SCL0, I²C) GPIO01	28
29	GPIO05		Ground	30
31	GPIO06		(PWM0) GPIO12	32
33	GPIO13 (PWM1)		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

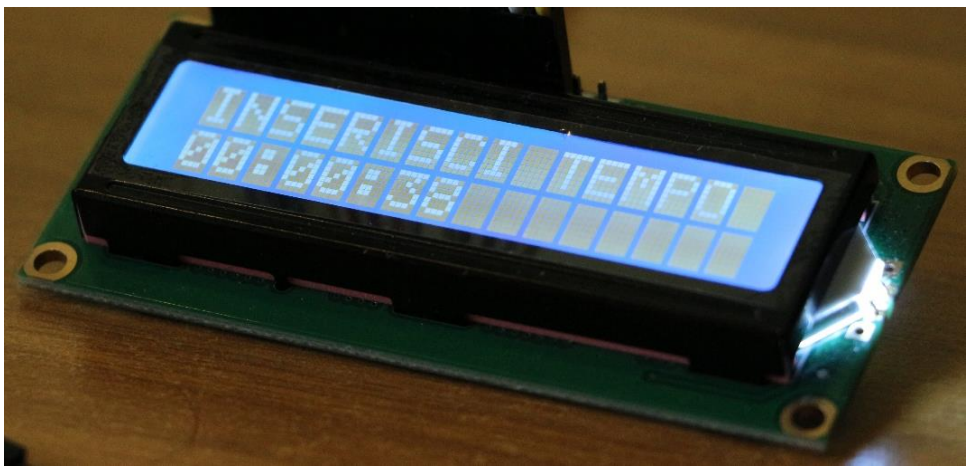
SCHERMO LCD 1602

CD1602 è un LCD a caratteri industriali in grado di visualizzare 16x02 o 32 caratteri contemporaneamente. Il principio del display LCD1602 è quello di utilizzare le caratteristiche fisiche del display a cristalli liquidi per controllare l'area di visualizzazione in base alla tensione.

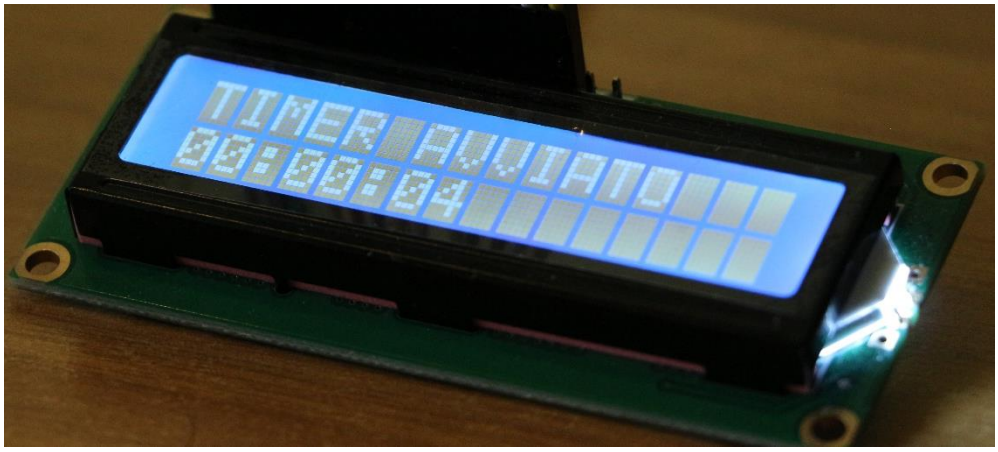
All'interno dello schermo verranno visualizzati gli stati in cui si trova il countdown di seguito descritte:

Inserisci tempo → stato iniziale in cui il countdown si trova all'avvio o dopo la pausa

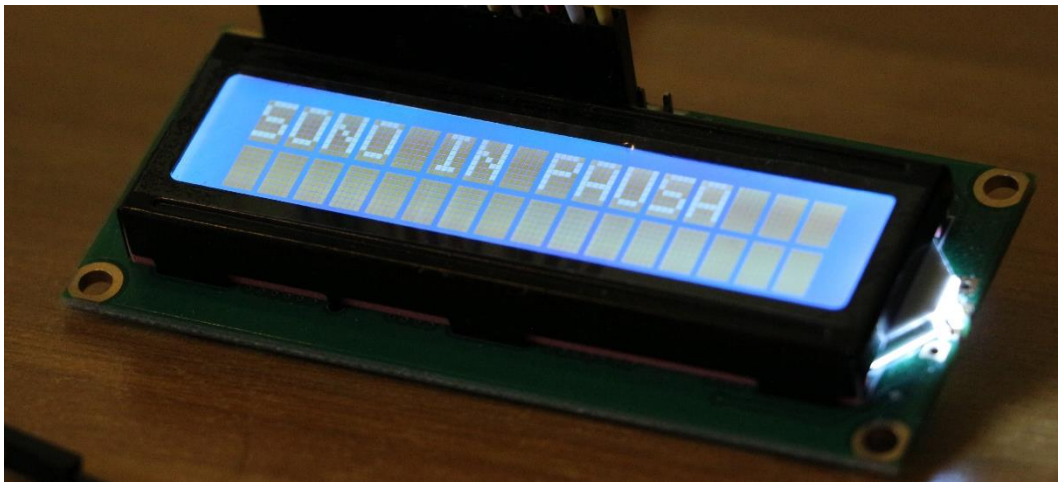
Inserisci tempo con secondi incrementati → quando l'utente setta i secondi



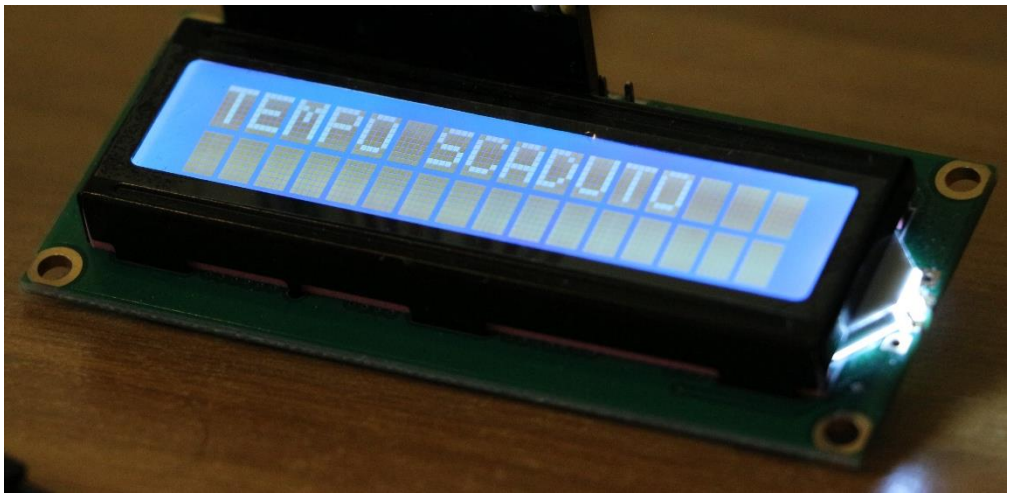
Timer avviato → in cui verrà mostrato il tempo che decorre



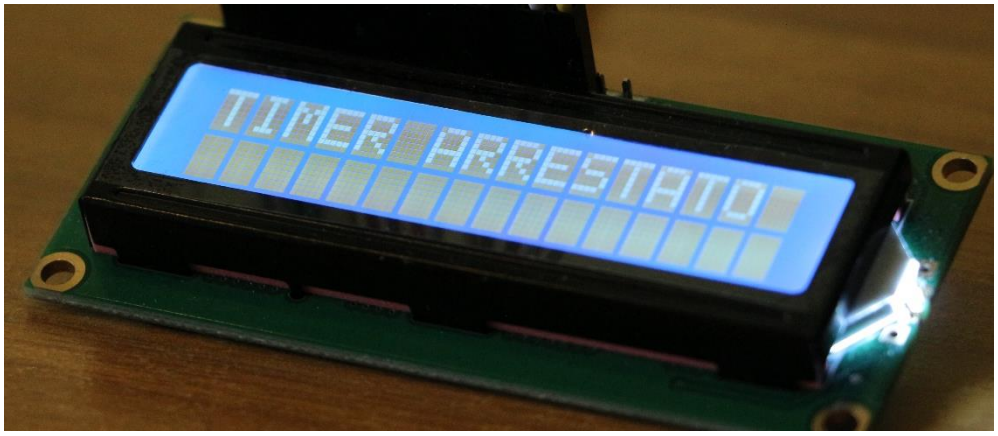
Sono in pausa → stato mostrato solo quando l'utente clicca sul pulsante di pausa



Tempo scaduto → stato mostrato quando il tempo è arrivato a zero

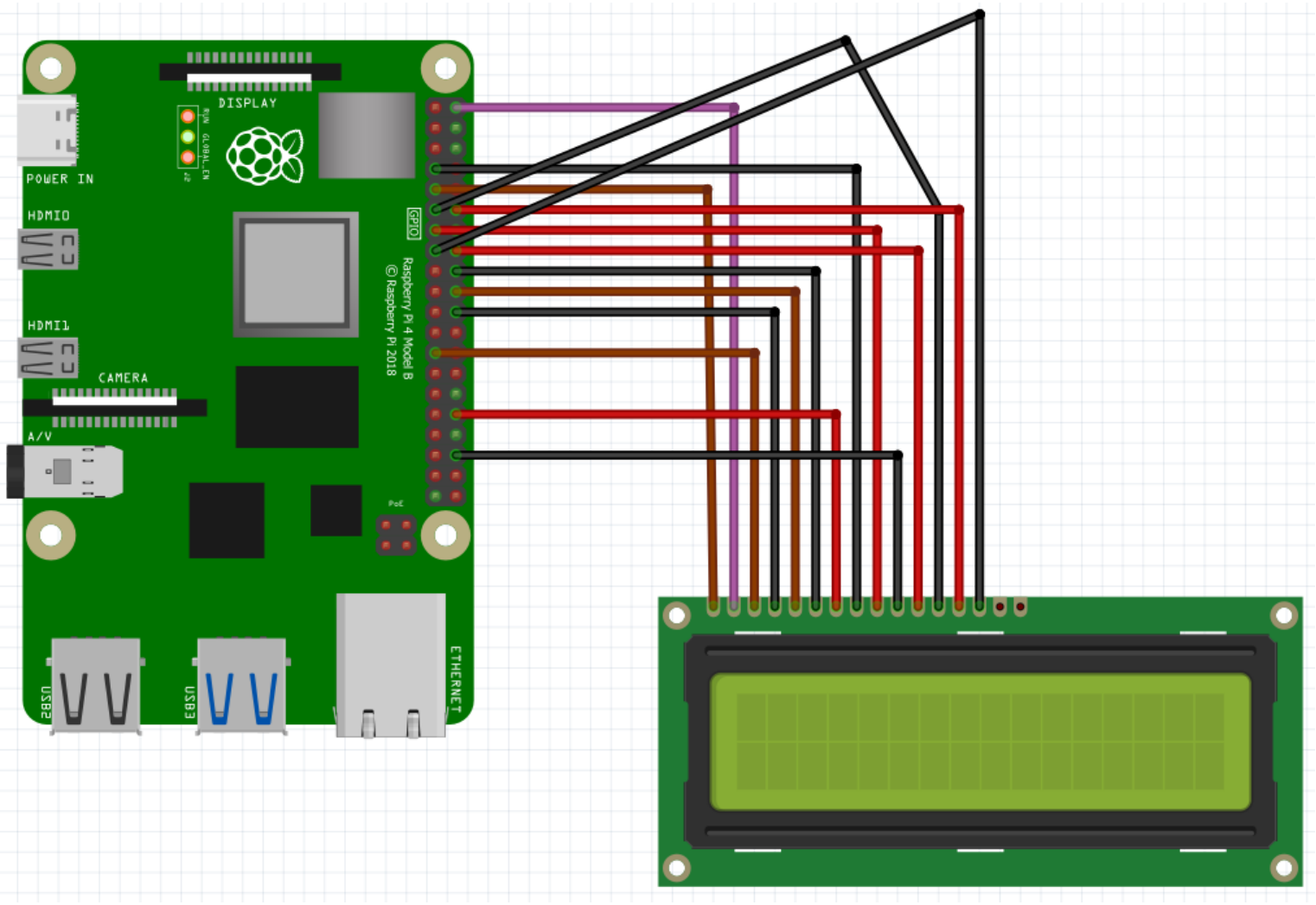


Timer arrestato → stato che indica l'arresto del countdown



Lo schermo LCD viene così collegato al target

LCD Pin	LCD Function	Raspberry Pin	Raspberry Function
01	VSS	09	Ground
02	VDD	02	5V
03	V0	25	Ground
04	RS	22	Gpio 25
05	RW	20	Ground
06	E	18	Gpio 24
07	D0	32	Gpio 12
08	D1	07	Gpio 04
09	D2	13	Gpio 27
10	D3	36	Gpio 16
11	D4	16	Gpio 23
12	D5	11	Gpio 17
13	D6	12	Gpio 18
14	D7	15	Gpio 22
15	A	Non collegato	
16	K	Non collegato	



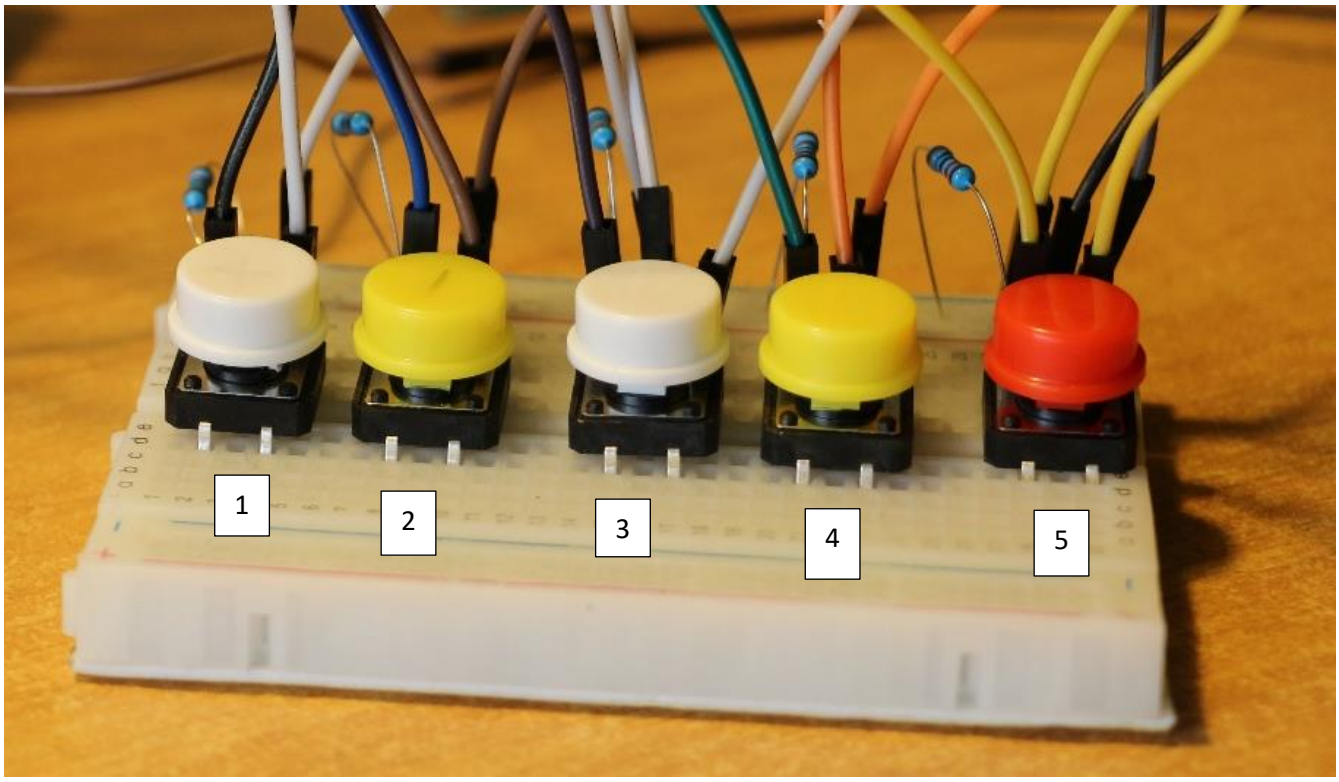
Descrizione funzioni LCD :

- VSS → pin collegato al + 5v per la retroilluminazione
- VDD → pin collegato a massa per la retroilluminazione
- V0 → pin dedicato per controllare e regolare il contrasto del display LCD
- RS → pin per la selezione del registro, indica al display dove memorizzare i dati inviati, collegandolo alla GPIO verranno memorizzati i dati nel registro dati.
- RW→ pin per la scrittura o la lettura nel display, collegandolo a massa impostiamo il display unicamente in modalità scrittura.
- E → pin che abilita la scrittura dei dati nei registri quando è pronto.
- D0 – D3 sono pin per la scrittura a 4 bit
- D4 – D7 sono pin per la scrittura a 8 bit
- A, K → Il pin A è collegato al + 5v per la retroilluminazione, mentre il pin K è collegato a massa per la retroilluminazione, entrambi non sono stati adoperati.

PULSANTI COLORATI

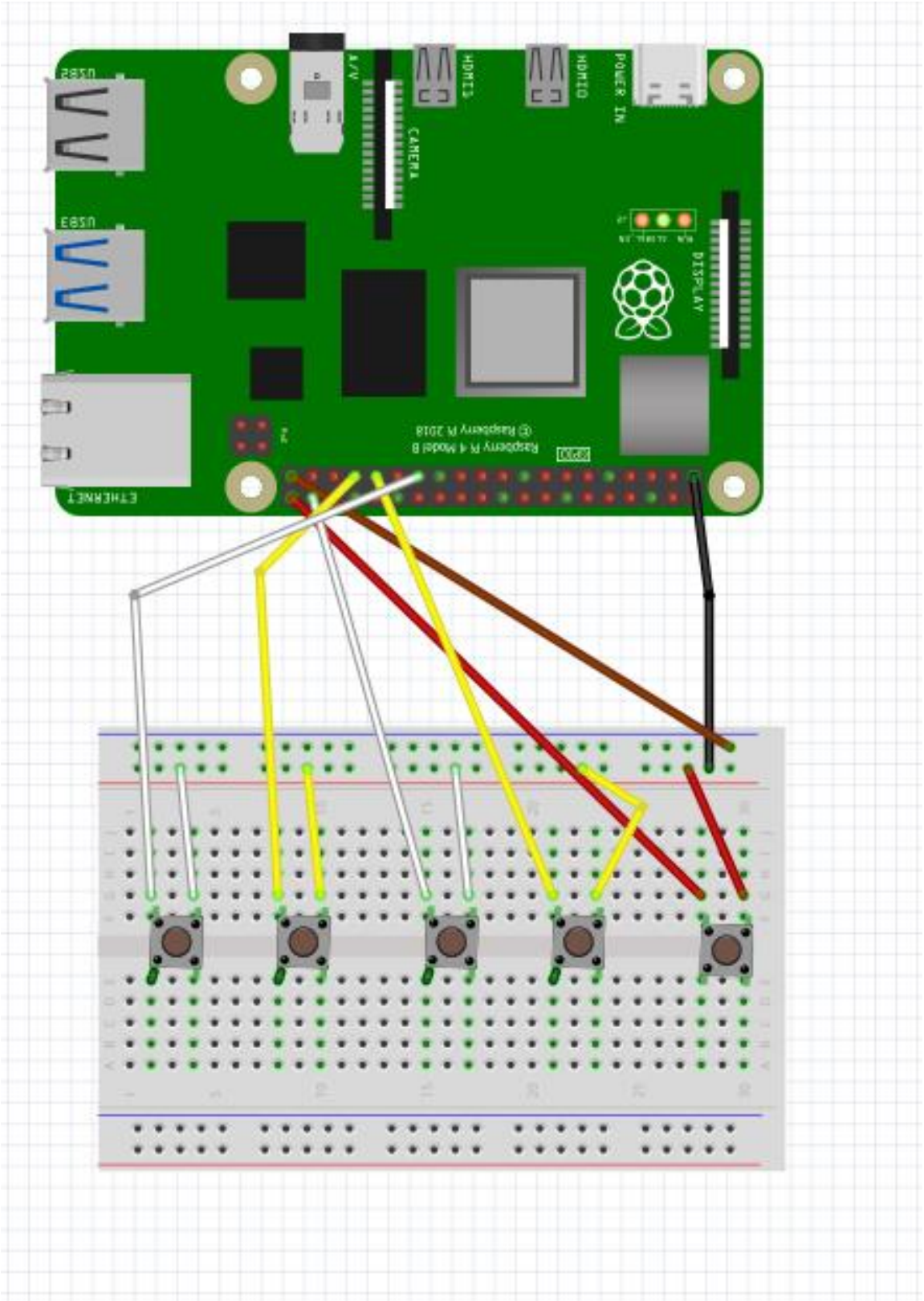
I pulsanti colorati nascono dall'esigenza di volere rendere il sistema più interattivo possibile con l'utente. Sono tutti posizionati nella mini-breadboard e le loro funzionalità sono:

1. Pulsante per incrementare il tempo (di colore bianco, il primo da sinistra), permette di incrementare i secondi;
2. Pulsante per decrementare il tempo (di colore giallo, il secondo da sinistra), permette di decrementare i secondi;
3. Pulsante per l'avvio (di colore bianco, al centro) , permette di avviare il countdown una volta che l'utente ha inserito la quantità di secondi desiderata, oppure per riavviare il countdown dopo che lo stesso si trova nello stato di pausa con lo scopo di far ripartire il conto alla rovescia da dove ;
4. Pulsante per la pausa (di colore giallo , il quarto da sinistra), permette di sospendere il countdown ogni qual volta che l'utente vuole metterlo momentaneamente in pausa;
5. Pulsante per lo stop (di colore rosso, il quinto da sinistra), questo pulsante ha una duplice funzione, permette infatti di bloccare il countdown in qualsiasi momento in modo tale da resettare il tempo. Permette anche di riavviare il countdown in questo modo il countdown ritornerà allo stato iniziale permettendo all'utente di reinserire un nuovo tempo per far partire un nuovo conto alla rovescia.



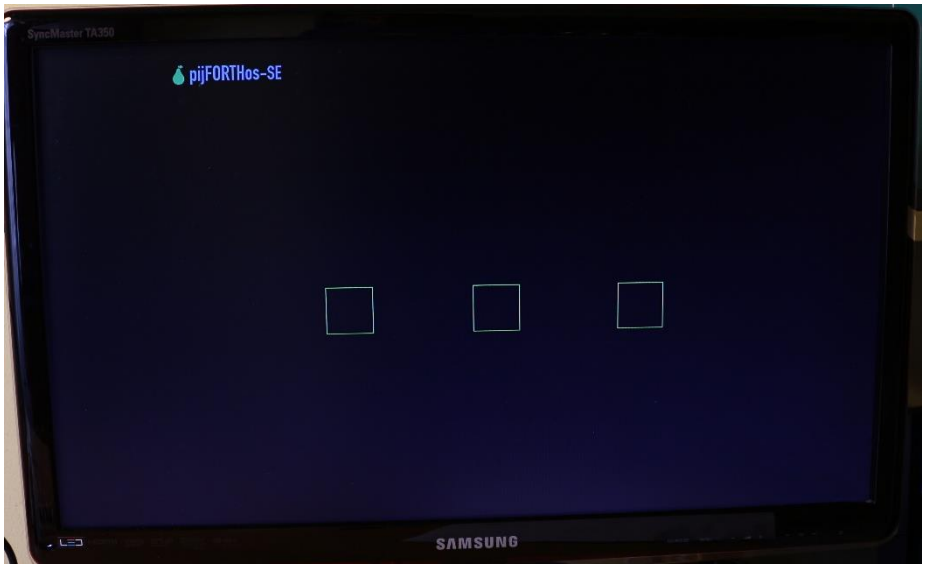
I pulsanti sono così collegati al target

Pulsante - funzionalità	Raspberry pin	Raspberry function
Bianco – Incremento Secondi	27	Gpio0
Giallo – Decremento Secondi	33	Gpio13
Bianco – Avvio/Start	38	Gpio20
Giallo – Pausa	31	Gpio6
Rosso – Stop/Reset	40	Gpio21



Come accennato precedentemente, l'uso dello schermo collegato in HDMI nasce dall'esigenza di avere una rappresentazione grafica dello status del countdown. All'avvio del sistema, a schermo verranno visualizzati, al centro tre quadrati con i bordi verdi che descriveranno lo stato in questo modo:

- Verranno visualizzati tutti i quadrati colorati di nero nei seguenti casi:
    - quando il countdown è stato avviato;
    - quando l'utente sta inserendo il tempo;
    - quando il countdown è stato riavviato e si ritorna nella fase di set del tempo.



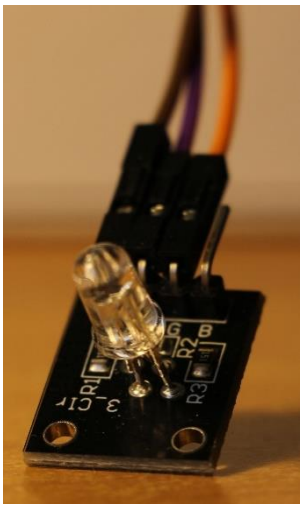
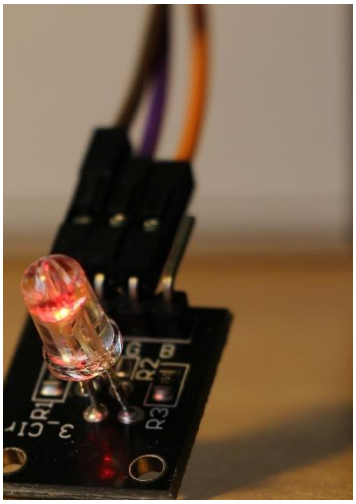
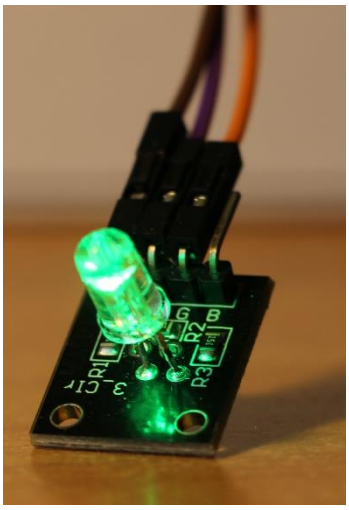
- Verrà visualizzato il primo quadrato colorato di bianco nel seguente caso:
    - quando l'utente pressa il pulsante bianco di avvio, in questo modo a schermo verrà visualizzato lo stato del countdown, ovvero che è stato avviato e che il tempo sta scorrendo;

- Verrà visualizzato il secondo quadrato colorato di giallo nel seguente caso:
    - quando l'utente pressa il pulsante giallo di pausa, in questo modo a schermo verrà visualizzato lo stato del countdown, ovvero che è stato messo in pausa;

- Verrà visualizzato il terzo quadrato colorato di rosso nel seguente caso:
    - Quando l'utente pressa il pulsante rosso di stop/reset, in questo modo a schermo verrà visualizzato lo stato del countdown, ovvero che è stato stoppato il tempo e sta per essere riavviato.

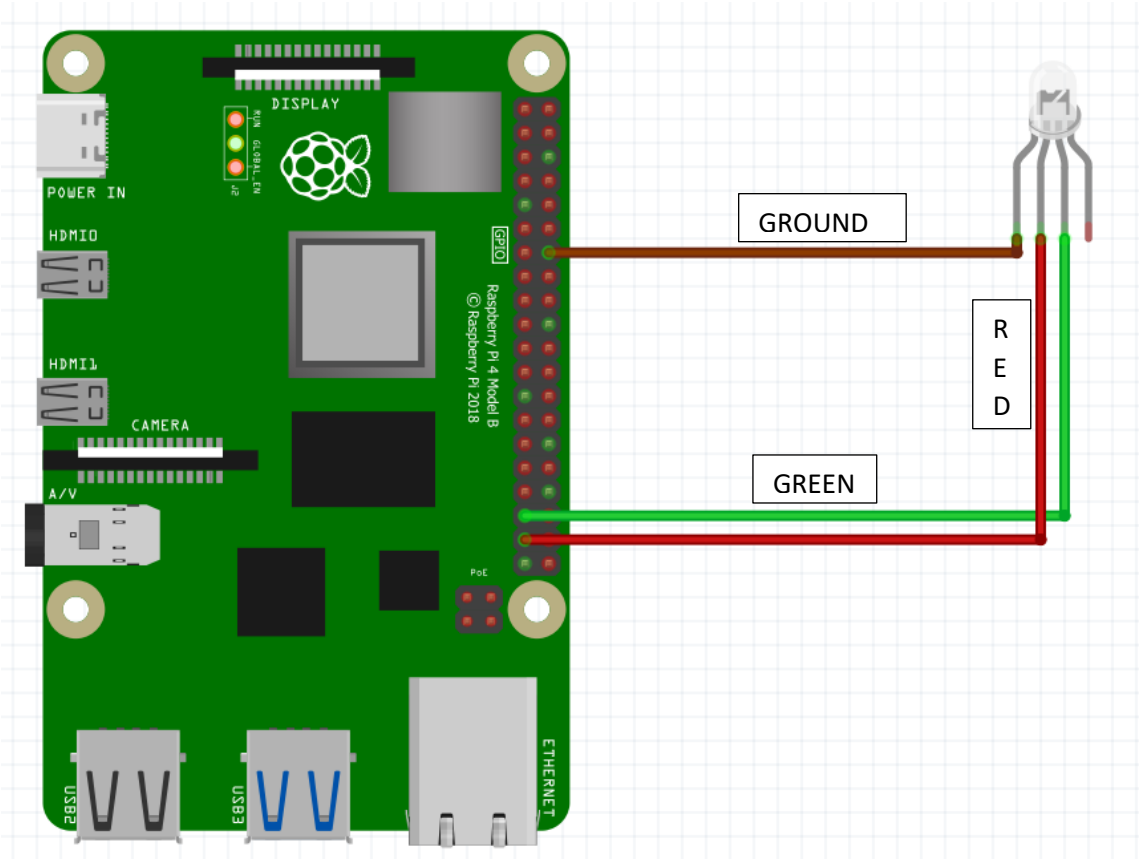
- Verranno visualizzati tutti i tre quadrati colorati nel seguente caso:
    - quando il countdown finisce. in questo modo a schermo verrà visualizzato lo stato del countdown, ovvero che il tempo è scaduto, e ci si aspetta che l'utente pressi il tasto stop per resettare il tempo e tornare nello stato di attesa di input.

La funzionalità del LED rgb è quella di descrivere attraverso le colorazioni dello stesso, il tempo residuo del countdown. Qualora mancassero 10 o meno secondi allo scadere del tempo il led lampeggerà di colore rosso, viceversa lampeggerà di verde.

Il led è così collegato al target

Canale LED	Raspberry pin	Raspberry function
Red	37	Gpio26
Green	35	Gpio19
Ground	14	Ground



CAPITOLO3 – DESCRIZIONE CODICE

In questa sezione ci sarà una breve descrizione del codice, scritto in Forth relativo alle implementazioni delle funzionalità per gestire : il countdown, le stampe in HDMI e LCD, e il funzionamento del Led.

Inoltre, nel codice è stata inserito il dizionario , in quanto alcune parole non venivano riconosciute.

INIZIALIZZAZIONE LCD

Lo schermo LCD contenente 16 pin connessi a quelli del PI, le GPIO interessate sono le seguenti:

```
DECIMAL

25 CONSTANT LCDRS
24 CONSTANT LCDE
12 CONSTANT LCD0
4  CONSTANT LCD1
27 CONSTANT LCD2
16 CONSTANT LCD3
23 CONSTANT LCD4
17 CONSTANT LCD5
18 CONSTANT LCD6
22 CONSTANT LCD7
```

Inoltre, per gestire le due righe dello schermo LCD sono state definite le seguenti costanti

```
\ identifica la prima riga del display indicata con 0
80 CONSTANT LCDLN1

\ identifica la seconda riga del display indicata con 1
C0 CONSTANT LCDLN2
```

Sfruttando la modalità a 8 bit il display viene inizializzato nel seguente modo:

```
\ Effettua l'inizializzazione dell'LCD settando le impostazioni desiderate.
: LCD-INIT
    LCD-SETUP

    38 LCDWRITE
    500 DELAY

    6 LCDWRITE
    500 DELAY

    C LCDWRITE
    500 DELAY

    LCDCLR
;
```

Le funzioni principali adoperate sono

```
\ Pulisce il display scrivendo il carattere SPAZIO su tutto l'LCD
\e riportando il puntatore all'inizio.
: LCDCLR 1 LCDWRITE ;


\ LCDNUMBER permette di scrivere numeri sul display, da destra verso sinistra
: LCDNUMBER LCDLN! LCDCURL<R LCDNTYPE ;


\ LCDSTRING permette di scrivere stringhe sul display, da sinistra verso destra
: LCDSTRING LCDLN! LCDCURL>R LCDSTYPE ;
```

Osservazione. Per stampare qualcosa sullo schermo va specificato "valore da stampare" "posizione" "riga";

- "Valore da stampare" è il numero/carattere. Il numero va direttamente scritto senza l'ausilio di istruzioni particolari mentre il carattere deve essere così indicato → S" mia_stringa."
- "Posizione" è la cella dello schermo da riempire con il valore sopra citato , lo schermo possiede 16 celle riempibili per riga;
- "Riga" è il numero di riga in cui si vuole stampare:
    - Con 0 → si indica la seconda riga
    - Con 1 → si indica la prima riga

GESTIONE DEI PULSANTI

Alla base del funzionamento di CountBerry vi è l'uso dei pulsanti. Come descritto precedentemente sono stati adoperati 5 diversi pulsanti , con lo scopo di gestire tutte le funzionalità desiderate per interagire con il sistema. Va osservato come, tutti i pulsanti collegati alle relative GPIO siano cablati a massa, ovvero tramite la breadboard il loro valore di default è 0. La funzionalità cardine che verifica se un pulsante è stato pressato o meno è la parola LEVEL:

```
\ ( GPIOn -- LEVn )
: LEVEL   1 SWAP LSHIFT GPLEV0 @ SWAP AND ;
```

I registri GPLEVn di General Purpose I /O (GPIO) vengono utilizzati per leggere i livelli sui pin configurati come ingressi digitali, come descritto dal commento una volta richiamata, tale parola, lascia nello stack un valore che sarà :

- 0 in quanto la GPIO è collegata a GND e abbiamo interpretato tale valore come tasto non è stato premuto.
- 0> cioè il valore è maggiore di 0 indica che il tasto è premuto, in quanto chiude il circuito e permette il passaggio di corrente.

Quindi con la scrittura GPIOX LEVEL sullo stack verrà inserito 0 se il pulsante non è pressato , un valore maggiore di zero viceversa. Alla base del funzionamento della Level vi è l'uso della costante GPLEV0 (con indirizzo FE200034) la quale ha come scopo quello di leggere lo stato dei pin.

Ogni Gpio adoperata viene quindi definita come costante

```
\GPIO0 COSTANTE RELATIVA ALLA GPIO COLLEGATA AL TASTO PER INCREMENTARE I SECONDI
0 CONSTANT GPIO0

\GPIO13 COSTANTE RELATIVA ALLA GPIO COLLEGATA AL TASTO PER DECREMENTARE I SECONDI
13 CONSTANT GPIO13

\GPIO6 COSTANTE RELATIVA ALLA GPIO COLLEGATA AL TASTO PAUSA
6 CONSTANT GPIO6

\GPIO21 COSTANTE RELATIVA ALLA GPIO COLLEGATA AL TASTO STOP/RESET
21 CONSTANT GPIO21

\GPIO20 COSTANTE RELATIVA ALLA GPIO COLLEGATA AL TASTO START/AVVIO
20 CONSTANT GPIO20
```

GESTIONE DEL TEMPO

Anche se il timer di sistema non è apparentemente legato all'ambiente esterno esso fornisce valori di input alla CPU. Nello spazio di indirizzamento ARM del Pi 4 B + con MMU disabilitata, l'indirizzo di base del timer di sistema è 0xFE003000.

Per misurare il tempo la CPU può semplicemente leggere l'ora di inizio dal registro CLO e quindi sfruttare un busy loop che legge il valore del registro corrente, sottrae il valore di inizio e confronta il risultato fino a quando quest'ultimo supera un determinato valore.

Tale comportamento è stato così implementato

```
HEX
\SYSCLO indica l’indirizzo del registro adoperato per gestire il tempo
FE003004 CONSTANT SYSCLO

\HALFSEC indica in esadecimale 500.000 usati per gestire i millisecondi
7A120 CONSTANT HALFSEC

\COMPARISON è una variabile usata per effettuare confronti e la inizializziamo a 0
VARIABLE COMPARISON
0 COMPARISON !

\INCREMENT permette di incrementare COMPARISON,
\sommando il valore del registro usato per il tempo con la variabile per gestitre i
 millisecondi
: INCREMENT SYSCLO @ HALFSEC + COMPARISON ! ;

\HALFSLEEP simula un attesa pari a mezzo secondo
: HALFSLEEP HEX INCREMENT BEGIN SYSCLO @ COMPARISON @ < WHILE REPEAT DROP DECIMAL ;
```

È stata prevista la parola LAYOUTSYS:

```
 : LAYOUTSIS 0 1 LCDSTRING ;
```

LAYOUTSIS quindi stampa le stringhe relative agli stati del countdown a partite della prima posizione a sinistra della prima riga dello schermo LCD.

Quindi in generale diremo che, nello schermo LCD verranno visualizzati gli stati del countdown con la parola indicata precedentemente; mentre i secondi settati dall'utente, e il tempo che scorre, verranno visualizzati con le parole qui di seguito definite

```
\TIME variabile in cui vengono immagazzinati i secondi settati dell'utente che
\viene settata a 0
VARIABLE TIME
: RESET 0 TIME ! ;
RESET

\Parole adoperate per indicare il punto in cui vanno visualizzati minuti e secondi
: PRINTMIN 4 0 LCDNUMBER ;
: PRINTSEC 7 0 LCDNUMBER ;

\Parole adoperate per visualizzare i ":" da preporre a minuti e secondi
: PRINTCOLONMIN HEX S" :" 2 0 LCDSTRING ;
: PRINTCOLONSEC HEX S" :" 5 0 LCDSTRING ;

DECIMAL
\Queste parole sfruttano l'aritmetica modulo 60 per visualizzare correttamente
\l'incremento e il decremento del tempo
: ZEROSEC TIME @ 60 /MOD SWAP 10 < IF 0 6 0 LCDNUMBER THEN ;

: ZEROMIN TIME @ 60 /MOD 600 < IF 0 3 0 LCDNUMBER THEN ;

: ZEROHOUR TIME @ 3600 < IF 0 0 0 LCDNUMBER 0 1 0 LCDNUMBER THEN ;

\Questa parola permette di stampare ore minuti e secondi con il loro separatori
: TIMER_INIT ZEROHOUR PRINTCOLONMIN ZEROMIN PRINTCOLONSEC ZEROSEC ;

\LYT sfruttando l'aritmetica modulo 60 stampa minuti e secondi
: LYT TIME @ 60 /MOD PRINTMIN PRINTSEC TIMER_INIT ;
```

Per il funzionamento del sistema è stato proposto un ciclo infinito con il seguente comportamento :

```
: GOON FIRST_BOX BEGIN GPIO6 LEVEL 0= GPIO21 LEVEL 0= AND WHILE BLINKLED  REPEAT ;

: COUNTDOWN BEGIN GPIO20 LEVEL 0> TIME @ 0> AND WHILE GOON PAUSA REPEAT ;

: TEST BEGIN TRUE WHILE SET_TIME STOP COUNTDOWN REPEAT ;

TEST
```

TEST permette di avviare CountBerry. Al suo interno sono richiamate le seguenti parole che verranno ripetutamente eseguite :

- SET_TIME, permette di visualizzare lo stato "Inserisci Tempo" e richiama le parole INCTIME e DECTIME:
    - INCTIME incrementa la variabile TIME di uno ogni qual volta si pressa il pulsante collegato alla GPIO0, e stampa a schermo il suo valore. La parola SLEEP permette di stampare correttamente l'incremento con una certa velocità di aggiornamento dei secondi.
    - DECTIME decrementa TIME di uno ogni qual volta si pressa il pulsante collegato alla GPIO13, e stampa a schermo il suo valore con valore di aggiornamento fornito da SLEEP.

```
: INCTIME GPIO0 LEVEL 0> IF TIME @ 1 + TIME ! LYT SLEEP THEN ;

: DECTIME GPIO13 LEVEL 0> TIME @ 0> AND IF TIME @ 1 - TIME ! LYT SLEEP THEN ;

: SET_TIME S" INSERISCI TEMPO" LAYOUTSIS INCTIME DECTIME ;
```

Osservazione su SLEEP. Il team durante l'implementazione dei pulsanti per il set del tempo, ha notato che servisse una parola per "rallentare" la velocità di incremento/decremento del tempo al click dei relativi pulsanti. Senza l'ausilio di una parola che eseguisse tale compito , al click dei pulsanti il countdown veniva incrementato/decrementato di mille valori al secondo.

Sfruttando il principio della HALFSLEEP sopra descritta , è stata proposta la parola SLEEP che permette appunto aggiornare l'incremento/decremento dei secondi sullo schermo LCD ogni 62500 millisecondi.

```
DECIMAL
\COMP1 è una variabile usata per effettuare confronti e la inizializziamo a 0
VARIABLE COMP1
0 COMP1 !

\AGGSEC indica in esadecimale 62.500 usata per attendere tot millisecondi prima di
\aggiornare il valore nello schermo
HEX
F424 CONSTANT AGGSEC

\INC permette di incrementare COMP1,sommando il valore del registro usato per il
tempo con la variabile per gestire i millisecondi
: INC SYSCLO @ AGGSEC + COMP1 ! ;

\SLEEP simula un attesa pari 1/16 di secondo
: SLEEP HEX INC BEGIN SYSCLO @ COMP1 @ < WHILE REPEAT DROP DECIMAL ;
```

- STOP, è la seconda parola inclusa nel ciclo principale, ha il compito di verificare se il pulsante collegato alla Gpio21 è stato pressato, qualora lo fosse :
    - stamperà "Timer Arrestato" sullo schermo LCD per 1,5 secondi e colorerà con la parola THIRD_BOX il terzo quadrato in HDMI di rosso (richiamante il colore del pulsante)
    - farà una clear dello schermo LCD
    - resetterà il countdown
    - aspetterà 1 secondo
    - stamperà in HDMI i 3 quadrati di colore nero

```
: STOP GPIO21 LEVEL 0> IF
   S" TIMER ARRESTATO" LAYOUTSIS THIRD_BOX 3HALFSLEEP LCDCLR RESET 2HALFSLEEP
   ALL_BLACK
   THEN ;
```

- COUNTDOWN, è la terza parola inclusa nel ciclo, si tratta anch'essa di un ciclo in cui

```
: COUNTDOWN BEGIN GPIO20 LEVEL 0> TIME @ 0> AND WHILE GOON PAUSA REPEAT ;
```

fino a quando il pulsante di avvio è stato pressato e il tempo è stato settato dall'utente esegue le parole GOON e PAUSA:

- GOON indica che il countdown sta eseguendo il conto alla rovescia, per tali ragioni con la parola FIRST_BOX colora il primo quadrato in HDMI di bianco (richiamante il colore del pulsante). Successivamente fin quando né il tasto di pausa , né il tasto di stop sono stati pressati eseguirà la parola blinkled.

```
: GOON FIRST_BOX BEGIN GPIO6 LEVEL 0= GPIO21 LEVEL 0= AND WHILE BLINKLED  REPEAT ;
```

Blinkled ha il compito di decrementare il tempo di un secondo e permette di far lampeggiare il led ogni secondo che passa dell'apposito colore.

```
: BLINKLED DECIMAL DECREMENT
   TIME @ 10 > IF LYT BLINKGREENLED S" TIMER AVVIATO" LAYOUTSIS
        ELSE TIME @ 0= IF RED LED OFF
        ELSE LYT BLINKREDLED S" TIMER AVVIATO" LAYOUTSIS THEN THEN ;
```

- DECREMENT quindi decrementa il countdown di un secondo, inoltre quando il tempo è uguale a zero, stamperà nello schermo LCD lo stato "Tempo scaduto" e colorerà tutti i quadrati in HDMI con la parola FINAL_BOX.

```
: DECREMENT TIME @ 0> IF TIME @ 1 - TIME ! ELSE S" TEMPO SCADUTO" LAYOUTSIS
        FINAL_BOX 3HALFSLEEP LCDCLR HALFSLEEP THEN DROP ;
```

- PAUSA invece, al click del pulsante collegato alla Gpio6 stamperà lo stato "Sono in pausa" nello schermo LCD fin quando il tasto viene pressato, quando lo stesso viene rilasciato, il ciclo principale continuerà il loop ritornando a SET-TIME in cui l'utente potrà far ripartire il countdown da dove lo aveva fermato, oppure incrementare o decrementare il countdown.
  Inoltre, al press del tasto pausa con la parola SECOND_BOX verrà colorato il secondo quadrato in HDMI di giallo (richiamante il colore del pulsante)

```
: PAUSA BEGIN GPIO6 LEVEL 0> WHILE S" SONO IN PAUSA" LAYOUTSIS SECOND_BOX
        3HALFSLEEP REPEAT ;
```

In HDMI sono stati rappresentati tre quadrati con bordi verdi , per indicare lo stato in cui il countdown si trova. I tre quadrati vengono appositamente colorati nel seguente modo:

- bianco se il countdown è avviato,
- giallo se è in pausa,
- rosso se è stato stoppato/resettato,
- tutti i quadrati colorati se il countdown è scaduto,
- nessun quadrato colorato, se il countdown è appena stato avviato o riavviato.

Il display HDMI del Pi è generato dal Videocore, un processore a bassa potenza, efficiente nel manipolare dati audio e video che ha il compito di generare il display HDMI.

Il display è quindi una matrice di pixel di larghezza * altezza, è possibile scegliere tali parametri per definire la risoluzione spaziale del display. Un pixel si trova attraverso le sue coordinate x = {0,..., width-1} e y = {0,..., height-1}. Ad esempio, il Pixel (0,0) è il più in alto a sinistra, il Pixel (larghezza-1,0) è il più a destra nella riga più in alto e il Pixel (larghezza-1, altezza-1) è l'ultima a destra in basso.

Il colore di un pixel è controllato da un valore codificato in bit, quando si utilizzano 32 bit per pixel (bpp), il colore del pixel viene codificato come valore ARGB: 8 bit per ciascuno dei componenti di colore blu e 8 bit per il canale alfa utilizzato per gli effetti di trasparenza.

I colori adoperati sono i seguenti, a causa di una somiglianza con i colori del Led sono stati adoperati alcuni nomi in italiano.

```
00FFFFFF CONSTANT WHITE
00000000 CONSTANT BLACK
00FF0000 CONSTANT ROSSO
00FFD700 CONSTANT YELLOW
007FFF00 CONSTANT VERDE
```

I valori dei pixel sono memorizzati in un framebuffer, che è una regione di memoria condivisa dal VC e dalla CPU, definito da noi con la costante → HEX 3E8FA000 CONSTANT FRAMEBUFFER

Il framebuffer è organizzato come una matrice di righe principali con valori di profondità di colore, i valori dei pixel vengono compressi uno dopo l'altro in un array lineare dei valori della profondità del colore dove, il primo elemento nell'array controlla il pixel più in alto a sinistra, in generale, il colore del pixel x, y è controllato dal valore i-esimo nell' array lineare, dove i = larghezza * y + x.

pijFORTHos-se configura il display HDMI per 1024x768x32bpp all'avvio. Il colore del pixel x, y è controllato dal valore ARGB nella parola all'offset di byte 4 * (1024 * y + x) dall'indirizzo di base del framebuffer, l'indirizzo di base viene così visualizzato nell'output all'avvio.

Osservando che la dimensione in byte di una linea di pixel (4 * 1024 = 4 * 0x400 = 0x1000, il "pitch") deve essere aggiunta all'indirizzo da cui far partire la linea, sono state definite le seguenti parole:

```
\per disegnare punti verso il basso,
: DOWN  2DUP ! 1000 + ;

\per disegnare punti verso destra.
: RIGHT  2DUP ! 4 + ;
```

Volendo disegnare delle linee orizzontali e verticali di n pixel abbiamo ripetuto la parole n volte quanto deve essere la lunghezza della linea.

Dovendo visualizzare a schermo tre quadrati sono state inoltre definite le seguenti parole per individuare il punto di partenza da cui disegnare le linee

```
: FIRST FRAMEBUFFER 100 4 * + 180 1000 * + ;
: SECOND FRAMEBUFFER 200 4 * + 180 1000 * + ;
: THIRD FRAMEBUFFER 300 4 * + 180 1000 * + ;
```

Inoltre, sperimentalmente volendo realizzare un quadrato di 51 pixel per lato sono state definite le seguenti parole:

```
: LINEADX
    BEGIN BOX_COUNTER @ 51 < WHILE +BOX_COUNTER RIGHT REPEAT 0BOX_COUNTER DROP DROP
;

: LINEADOWN
    BEGIN BOX_COUNTER @ 51 < WHILE +BOX_COUNTER DOWN REPEAT 0BOX_COUNTER DROP DROP
;
```

Il funzionamento è il seguente: sfruttando il contatore BOX_COUNTER inizializzato a zero, fin quando lo stesso è minore di 51 verrà ripetuta l'istruzione per disegnare una linea verso destra/verso il basso. Essendo tale contatore riutilizzato, alla fine del ciclo verrà riinizializzato a zero.

Ogni linea del quadrato verrà disegnata specificando i seguenti parametri:

"colore" "posizione_di partenza" "orientamento_linea"

```
: BOX_INIT_FIRST
    VERDE FIRST LINEADX
    VERDE FIRST 50 1000 * + LINEADX
    VERDE FIRST 51 4 * + LINEADOWN
    VERDE FIRST LINEADOWN
;
```

Tale comportamento implementato per ogni quadrato viene racchiuso dentro la seguente istruzione, così all'avvio del sistema, in HDMI verranno visualizzati tre quadrati.

```
: BOX_INIT
    BOX_INIT_FIRST
    BOX_INIT_SECOND
    BOX_INIT_THIRD
;

BOX_INIT
```

Infine volendo colorare un quadrato è stata definita la parola Colora, il cui scopo è quello di disegnare delle linee orizzontali "simulando" così l'azione di colorare

```
: COLORA
    BEGIN BOX_COUNTER @ 50 < WHILE
    +BOX_COUNTER
    RIGHT
    REPEAT 0BOX_COUNTER DROP DROP
```

"Colora" viene richiamata all'interno delle apposite parole per colorare i quadrati nel modo opportuno

```
: BOX_FIRST_WHITE
    WHITE FIRST 4 1000 + + COLORA
    BEGIN NLINE @ 50 <
    WHILE
        WHITE FIRST 4 + 1000 NLINE @ * + COLORA
        +NLINE
    REPEAT
    1 NLINE !
;
```

Per colorare gli appositi quadrati al click di uno specifico pulsante sono state definite le seguenti parole richiamate rispettivamente nelle apposite istruzioni relative alla gestione del tempo.

```
: FIRST_BOX
    BOX_FIRST_WHITE
    BOX_SECOND_BLACK
    BOX_THIRD_BLACK
;

: SECOND_BOX
    BOX_FIRST_BLACK
    BOX_SECOND_YELLOW
    BOX_THIRD_BLACK
;

: FINAL_BOX
    BOX_FIRST_WHITE
    BOX_SECOND_YELLOW
    BOX_THIRD_ROSSO
;

: THIRD_BOX
    BOX_FIRST_BLACK
    BOX_SECOND_BLACK
    BOX_THIRD_ROSSO
;

: ALL_BLACK
    BOX_FIRST_BLACK
    BOX_SECOND_BLACK
    BOX_THIRD_BLACK
;
```

LED

La funzionalità implementata con il led RGB è la seguente:

- Led lampeggia di verde ogni secondo di default
- Led lampeggia di rosso quando mancano 10 secondi allo scadere del countdown

Per gestire il countdown sono state dichiarate le seguenti costanti

```
HEX
FE200008 CONSTANT GPFSEL2
FE200004 CONSTANT GPFSEL1
FE20001C CONSTANT GPSET0
FE200028 CONSTANT GPCLR0
```

Come descritto precedentemente il canale del led verde è collegato alla Gpio19 mentre il canale rosso è collegato alla Gpio26. Entrambe le Gpio di default sono impostate a "low" quindi bisogna settare in output i bit relativi ad esse. Osservando che la Gpio19 (bit 29-27) viene gestita dalla GPFSEL1 (indirizzo FE200004) mentre la Gpio26 (bit 20-18) dalla GPFSEL2 (indirizzo FE200008), abbiamo settato entrambe le Gpio in output (bit 001) definendo le apposite bitmask;

```
08224000 GPFSEL1 !
12040000 GPFSEL2 !
```

Quindi per accendere o spegnere il LED è possibile utilizzare i valori lasciati nello stack dal LED rimuovendo l'indirizzo di registro indesiderato per ogni caso, nello specifico:

- Per accendere il LED, il valore TOS (indirizzo di GPCLR0) è DROPped , così è stata definita la parola ON come → : ON DROP ! ;
- Mentre per spegnere il LED, il valore TOS è NIPped (NIP può essere definito come SWAP DROP), così è stata definita la parola OFF come → : OFF NIP ! ;

Il colore rosso essendo collegato alla Gpio26 è identificato nel seguente modo → : RED 04000000 ;

Mentre il colore verde essendo collegato alla Gpio19 è identificato nel seguente modo → : GREEN 80000 ;

Infine, il Led può essere definito come : LED GPSET0 GPCLR0.

Per far lampeggiare il Led ogni secondo e per sincronizzare il lampeggio del Led con quello dello schermo LCD sono state usate due parole (una per ogni colore implementato) così descritte:

```
: BLINKGREENLED GREEN LED ON HALFSLEEP GREEN LED OFF LCDCLR HALFSLEEP ;
: BLINKREDLED RED LED ON HALFSLEEP RED LED OFF LCDCLR HALFSLEEP ;
```

La parola HALFSLEEP descritta precedentemente simula l'attesa di mezzo secondo, mentre la parola LCDCLR permette la pulizia dello schermo così da simulare il blink dello stesso.

Queste due parole sono richiamate all'interno della parola BLINKLED, che ha un duplice scopo; la prima è quella di decrementare ogni secondo il countdown, la seconda è quella di far lampeggiare il led nel seguente modo:

```
: BLINKLED DECIMAL DECREMENT
    TIME @ 10 > IF LYT BLINKGREENLED S" TIMER AVVIATO" LAYOUTSIS
         ELSE TIME @ 0= IF RED LED OFF
                ELSE LYT BLINKREDLED S" TIMER AVVIATO" LAYOUTSIS
            THEN THEN ;
```

CAPITOLO4 – CASI D'USO

Di seguito riportati quattro casi d'uso al fine di mostrare il comportamento del sistema in quattro situazioni diverse. Si consiglia di visualizzare i video in 1080p HD.

CASO D'USO 1

Il primo caso d'uso mira ad osservare uno scenario tipico di avvio del countdown senza interruzioni del tempo, con la seguente sequenza di azioni:

1. Fase di "Inserisci Tempo" in cui l'utilizzatore setta i secondi desiderati con gli appositi pulsanti
2. Fase di "Timer Avviato" quando l'utilizzatore presserà il tasto di start/avvio. Al press del pulsante nello schermo LCD verranno visualizzati i secondi che scorrono , mentre in HDMI verrà colorato il primo quadrato a sinistra di bianco, che indica che il countdown è stato avviato.
3. Fase di "Tempo Scaduto" che indica tramite una stringa sullo schermo LCD che il countdown ha terminato, inoltre in HDMI verranno visualizzati tutti e tre i quadrati colorati
4. Soltanto quando l'utilizzatore presserà il tasto di reset, sullo schermo LCD verrà visualizzato lo stato di "Timer Arrestato" e in HDMI verrà visualizzato solo l'ultimo quadrato colorato in rosso. Successivamente si ritorna allo stato di "Inserisci Tempo" con tutti i tre quadrati di colore nero.

NB: Come descritto precedentemente quando mancheranno meno di 10 secondi al termine il led lampeggerà di rosso anziché di verde.

Video → caso uso 01.mp4

CASO D'USO 2 e 3

Il secondo caso d'uso mostra uno scenario in cui l'utilizzatore dopo aver avviato il countdown decide di azionare il pulsante di pausa, la sequenza di azioni è la seguente:

1. Dopo la fase di avvio il tempo sta per scorrere, e il countdown si trova nella fase di "Timer Avviato"
2. L'utilizzatore al click del pulsante di pausa imposterà il countdown nello stato "Sono in Pausa"
    a. Nel caso d'uso due possiamo vedere come l'utente non vuole modificare il tempo e ricomincia subito il conto alla rovescia cliccando il tasto play
    b. Nel caso d'uso tre possiamo vedere come l'utente vuole modificare i secondi, attraverso gli appositi pulsanti e successivamente ricomincia con il conto alla rovescia

    In entrambi gli scenari al press del tasto pausa, in HDMI il quadrato centrale viene colorato di giallo, mentre nello schermo LCD , verrà visualizzato lo stato "Sono in pausa per qualche secondo , per poi ritornare subito alla fase di "Inserisci tempo" dove l'utente può scegliere se modificare il tempo oppure farlo ripartire esattamente da dove l'ha stoppato.
3. Segue la sequenza di azioni descritta precedentemente relativa allo scorrere del tempo.

Video 2 → caso uso 02.mp4
Video 3 → caso uso 03.mp4

CASO D'USO 4

L'ultimo caso d'uso mostra uno scenario in cui l'utilizzatore dopo aver avviato il countdown decide di azionare il pulsante di stop, la sequenza di azioni è la seguente

1. Dopo la fase di avvio il tempo sta per scorrere, e il countdown si trova nella fase di "Timer Avviato"
2. L'utilizzatore clicca sul pulsante di stop/reset, il countdown entra nello stato di "Timer Arrestato" che verrà visualizzato nello schermo LCD, mentre in HDMI viene colorato l'ultimo quadrato di rosso. Questo stato è di tipo transitorio, in quanto ha lo scopo di riportare il countdown alla fase iniziale di "Inserisci Tempo" con tutti i quadrati non colorati mostrati in HDMI.
3. L'utilizzatore, quindi, potrà inizializzare un nuovo countdown e verrà eseguita la sequenza di azioni descritta nei punti precedenti.

Video 4 → caso uso 04.mp4

CAPITOLO5 –CONCLUSIONI

Il progetto qui presentato ha soddisfatto pienamente le richieste e l'idea da cui il team è partito, soprattutto per quanto riguardo l'indipendenza, in quanto non vi il bisogno di un computer per avviare il countdown. Le simulazioni effettuate hanno dato esito positivo in quanto il sistema riesce ad assolvere correttamente il proprio compito , senza bug o malfunzionamenti. Il tutto è stato reso in modo più interattivo possibile tramite l'ausilio dei pulsanti, ma anche tramite gli schermi per notificare all'utente lo stato del countdown. L'approccio adoperato dal team per lo sviluppo del progetto è stato la scrittura del codice in maniera modulare, grazie anche al linguaggio Forth che permette di verificare la correttezza di ogni istruzione ad ogni passo. Partendo dalle indicazioni fornite durante lo svolgimento del corso, il team è riuscito a sfruttare le potenzialità del sistema target per:

1. Gestione del tempo → attraverso il System Timer , il team è riuscito correttamente a gestire i secondi, sfruttando un apposito registro denominato SYSCLO con indirizzo 0xFE003000.
2. Gestione dei pulsanti → il monitoraggio dei livelli di voltaggio dei vari pulsanti è avvenuta attraverso la funzione level. Tale funzionalità verifica il livello della GPIO corrispondente al pulsante ogni qual volta che lo stesso viene pressato.
3. HDMI → pijFORTHos-se configura l'inizializzazione del display HDMI per 1024x768x32bpp allo start, il team sfruttando apposite le nozioni apprese durante il corso , è riuscito a gestire correttamente la manipolazione dei pixel per ottenere disegni a schermo
4. LCD → consultando documentazioni on line/ git hub il team è riuscito a gestire le stampe nello schermo LCD 1602

Nonostante ciò, sono state riscontrate alcune criticità

1. Sensibilità dei pulsanti nei tasti di pausa e stop, in quanto vanno pressati a fondo per eseguire correttamente la propria funzionalità.
2. Regolazione del contrasto e illuminazione dello schermo LCD, in quanto sebbene lo schermo sia dotato di potenziometro integrato lo stesso non è adatto per rendere più visibili le stampe.

In conclusione, si prevedono delle implementazioni future per realizzare una versione più costosa del progetto appena descritto e più funzionale. Il team si prefissa le seguenti migliorie:

- Miglioramento della pressione dei tasti, sfruttando anche dei tasti touchscreen;
- Visualizzazione migliore dello schermo, gestendo un potenziometro esterno;
- Applicazione sonora per avvisare l'utente quando il tempo si esaurisce.