



UNIVERSITÀ DEGLI STUDI DI SALERNO

LAUREA TRIENNALE

CORSO DI LAUREA IN INGEGNERIA ELETTRONICA

ESAME DI

Misure Per L' Automazione

**Manuale Utente scheda NUCLEO-F401RE + LCD 16x2 + Termistore +
Active Buzzer**

Professore

Vincenzo Paciello

Studente

Russo Luca

Anno Accademico 2020-2021

Indice

1. Introduzione al progetto
2. Comunicazione seriale RS232
3. Interfaccia
4. Frequenza di clock del processore
5. Real Time Clock
6. Termistore
7. LCD 16x2
8. Tasto USER (BLU)
9. LED
10. Active Buzzer
11. Criticità dell'attuale implementazione
12. Future implementazioni

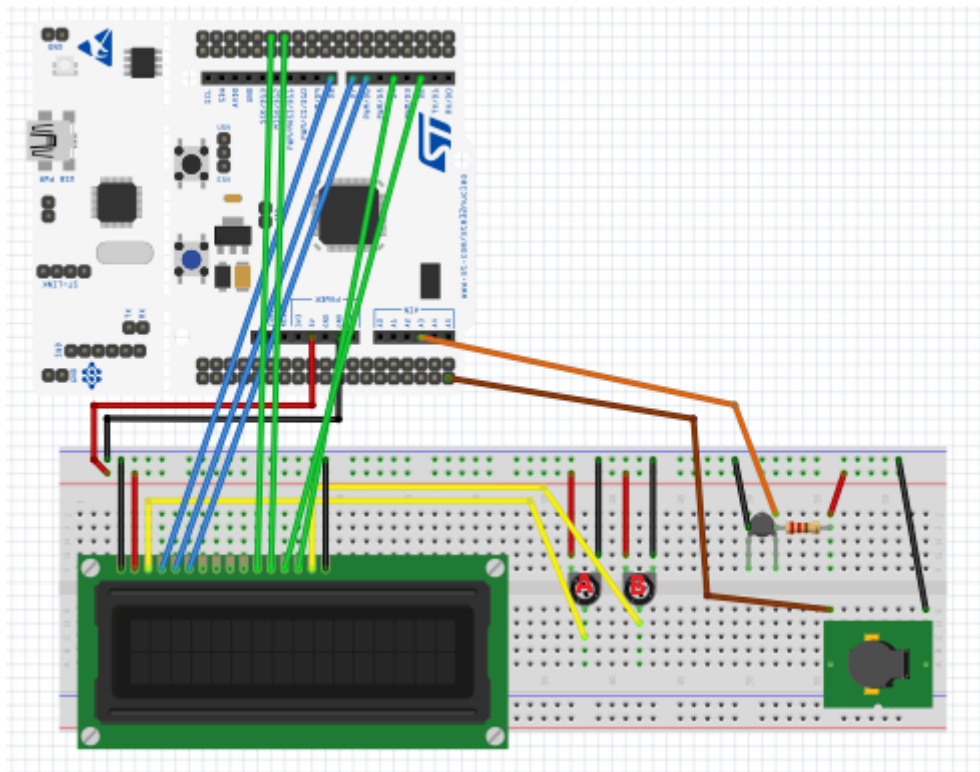
1. Introduzione al progetto

Il seguente progetto è stato sviluppato per rispondere ai quesiti proposti nell'ambito dell'esame di "Misure Per L' Automazione", tenuto dal professore Vincenzo Paciello presso l'Università degli Studi di Salerno. La traccia d'esame propostaci era la seguente:

1. Variare la frequenza del processore utilizzando il moltiplicatore interno
2. Comunicazione RS232 con il PC tramite un terminale (nel nostro caso Hercules)
3. Realizzare un menù di selezione per l'utente
4. Configurare il RTC
5. Realizzare le funzioni di lettura e settaggio del RTC
6. Acquisire la temperatura dal sensore di temperatura interno al microcontrollore
7. Applicare al segnale in uscita dal sensore di temperatura filtraggio
8. Utilizzare il tasto blu della scheda Nucleo per interagire con il programma realizzato
9. Utilizzare il led (variando la frequenza di lampeggio)

Vista la libertà di progetto concessa, è stato modificato il punto 6 leggendo la temperatura, piuttosto che dal sensore di temperatura integrato sulla scheda, tramite un termistore esterno. Inoltre, è stato utilizzato uno schermo LCD 16x2 ed un buzzer. L'implementazione di tutti i dispositivi utilizzati sarà trattata nei seguenti paragrafi. Tutto il software è stato scritto in linguaggio C, senza l'utilizzo di alcuna libreria, utilizzando come ambiente di sviluppo "Keil μ Vision5".

Di seguito è illustrato lo schema dei collegamenti effettuato tra tutti i componenti.



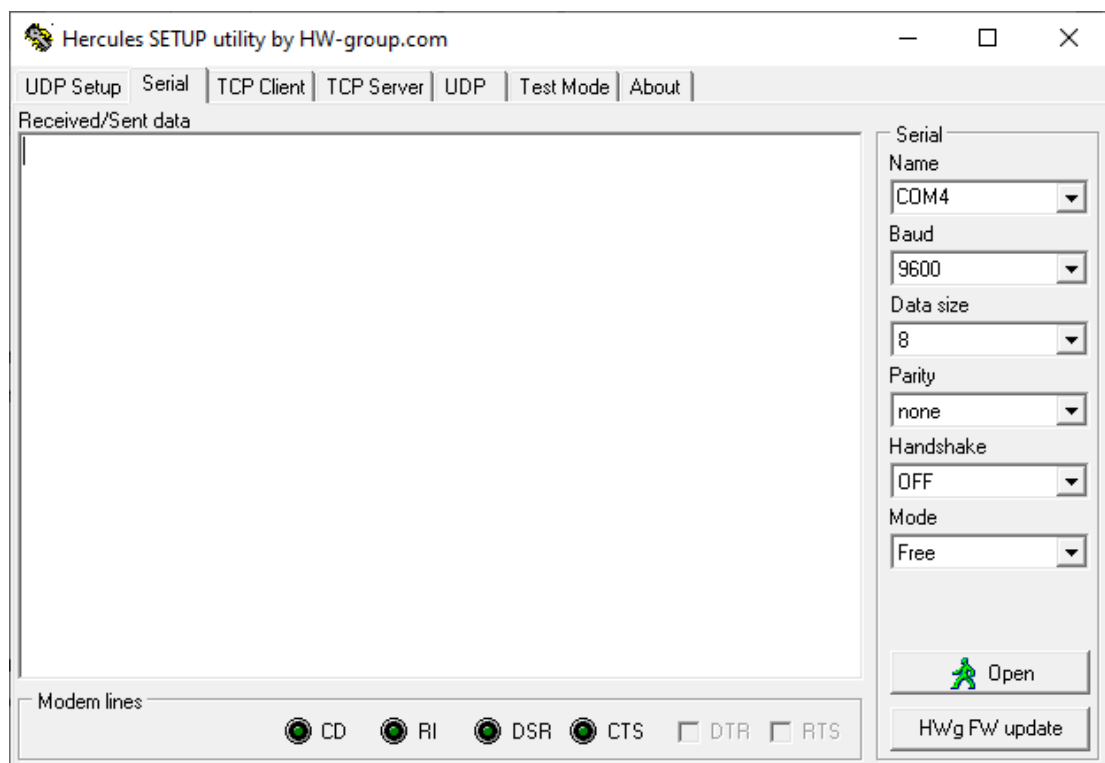
(Fig. 1 Schema collegamenti tra NUCLEO-F401RE e periferiche)

2. Comunicazione seriale RS232

La scheda NUCLEO-F401RE è stata programmata per poter dialogare con il pc tramite comunicazione seriale RS232. In questo caso, è stata implementata una connessione di tipo UART (Universal Asynchronous Receiver Transmitter) con le seguenti caratteristiche:

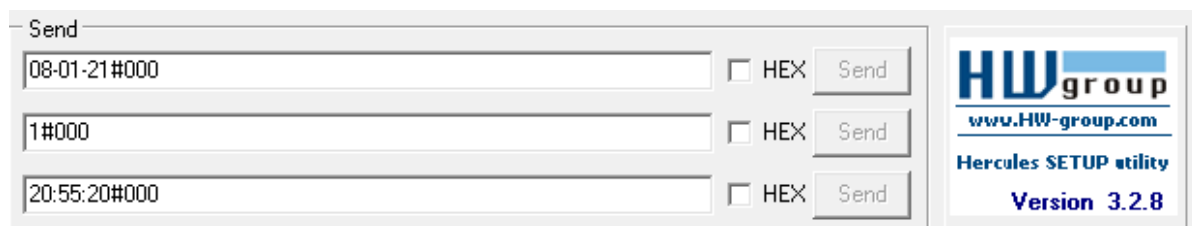
- Baud Rate: 9600Kbps
- Data Size: 8 bit
- Parity: none
- Handshake: OFF
- Mode: Free

Questi parametri dovranno essere settati sul terminale che dovrà comunicare con il microcontrollore. Di seguito un esempio su terminale "Hercules":



(Fig. 2 Settaggi per Seriale RS232 su Hercules)

Assicurarsi, per il corretto funzionamento, che venga sempre aggiunto il carattere di fine stringa '\0'. Sempre facendo riferimento all' esempio precedente, dal momento che Hercules non aggiunge in automatico il carattere '\0' in coda alla stringa, dobbiamo farlo noi manualmente scrivendo '#000'.



(Fig. 3 Invio stringhe tramite Hercules)

Essendo la porta seriale configurata nel modo più semplice possibile, senza controllo sulla parità (cioè se il numero di bit settati ad 1 è pari o dispari), e senza controllo hardware tramite i bit RTS e CTS (indicano rispettivamente la disponibilità a Ricevere e Trasmettere dati), potrebbero, a volte, presentarsi errori di comunicazioni. Molti di questi sono stati gestiti dal software, ma potrebbe presentarsi comunque l'eventualità che un'operazione debba essere ripetuta.

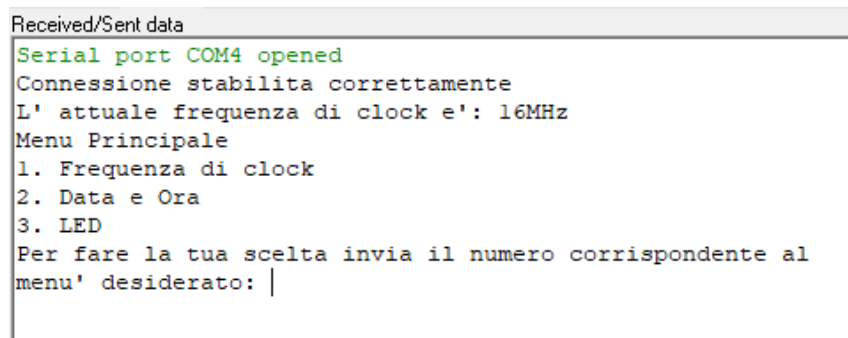
3. Interfaccia

All'accensione del microcontrollore, oltre all'accensione dell'LCD e all'avvio della rilevazione della temperatura che saranno approfondite nei paragrafi successivi, sarà stampato sul terminale connesso tramite porta seriale:

1. La scritta "Connessione stabilita Correttamente"
2. Informazioni sull'attuale frequenza di clock
3. Il menu principale

Quest'ultimo permetterà all'utente di scegliere tra tre opzioni:

1. Cambiare la frequenza di clock
2. Cambiare la data e ora (che vede stampate sul display)
3. Far Lampeggiare il LED2



```
Received/Sent data
Serial port COM4 opened
Connessione stabilita correttamente
L' attuale frequenza di clock e': 16MHz
Menu Principale
1. Frequenza di clock
2. Data e Ora
3. LED
Per fare la tua scelta invia il numero corrispondente al
menu' desiderato: |
```

(Fig. 4 Menu Principale)

Il software resterà in attesa che l'utente faccia la sua scelta e in base alla scelta effettuata invierà tramite seriale il menu corrispondente. Avremmo quindi tre menu:

1. Menu frequenza
2. Menu Data/Ora
3. Menu Led

In [menu frequenza](#) l'utente avrà la possibilità di scegliere tra quattro frequenze di clock possibili:

1. 16MHz
2. 24MHz
3. 32MHz

In [menu Data/Ora](#) l'utente potrà scegliere se cambiare la data e l'ora corrente

In [menu Led](#) l'utente potrà scegliere tra tre modalità di funzionamento per il led2 integrato sulla scheda:

1. Led ON
2. Led OFF
3. Lampeggia LED

Scegliendo la modalità 3, sarà chiesto all'utente di scegliere con quale frequenza il led dovrà lampeggiare. L'utente dovrà inserire un numero intero compreso tra 1 e 100 che indicherà

periodo della frequenza di lampeggiamento. Ad esempio, inserendo 1 secondo il Led risulterà acceso per 0.5 secondi e per 0.5 secondi acceso.

```
Menu frequenza
Scegli tra le seguenti frequenze:
1. 16MHz
2. 24MHz
3. 32MHZ
Per fare la tua scelta invia il numero corrispondente
alla frequenza desiderata: |
```

(Fig. 5 Menu frequenza)

```
Menu Data/Ora
1. Cambia Data/Ora
2. Torna al menu' principale
Per fare la tua scelta invia il numero corrispondente
alla funzione desiderata: |
```

(Fig.6 Menu Data/Ora)

```
Menu Led
Scegli tra le seguenti modalita' di funzionamento per il
Led:
1. Led acceso fisso
2. Led spento
3. Led lampeggiante
Per fare la tua scelta invia il numero corrispondente
alla funzione desiderata: |
```

(Fig. 7 Menu LED)

4. Frequenza di clock del processore

Il microcontrollore in oggetto può avere più sorgenti di clock, come indicato nel datasheet, che sono:

1. HSI oscillator clock
2. HSE oscillator clock
3. Main PLL (PLL) clock
4. LSI RC
5. LSE

In questo caso le sorgenti di clock che sono state utilizzate in tutto il progetto sono:

1. HSI
2. PLL
3. LSI RC

Le prime due sono utilizzate come sorgenti di clock per l'intero sistema, mentre l'ultima viene utilizzata come sorgente di clock unicamente per il Real Time Clock.

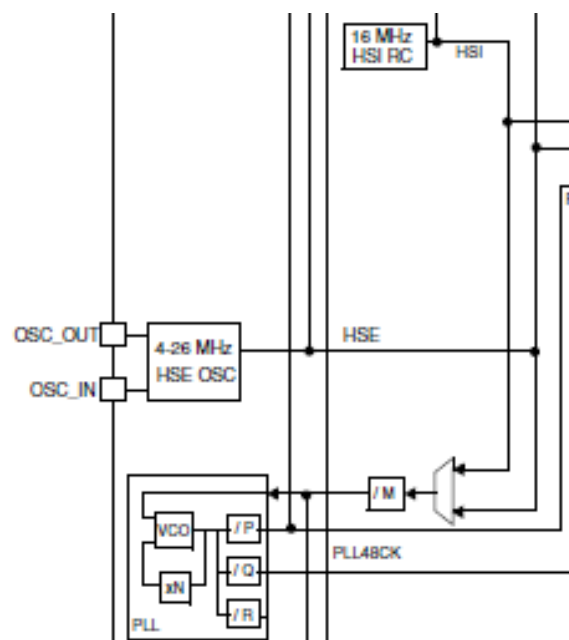
In questo paragrafo ci soffermeremo ad analizzare le prime due, mentre la trattazione dell'LSI sarà posposta al paragrafo relativo al funzionamento del RTC.

Il segnale di clock dell'HSI è generato internamente da un oscillatore a 16MHz ed è possibile utilizzarlo sia direttamente come system clock che come ingresso del PLL.

Come è possibile vedere dal seguente schema, estratto dal datasheet "RM0368_stm32f401xbc-and-stm32f401xde", il PLL può prendere come ingresso sia il segnale dell'HSI che del HSE (selezionabile tramite il multiplexer in figura). Poi settando in modo adeguato i registri P, N ed M è possibile modificare in segnale in ingresso generando in uscita il segnale desiderato.

La formula utilizzata è la seguente:

$$f_{pll} = \left(f_{input} * \frac{N}{M} \right) * \frac{1}{P}$$



(Fig. 8 Schema PLL)

Quindi, in base alle scelte effettuate dall'utente, il software gestirà sia la selezione della sorgente di clock per l'intero sistema (gestendo un multiplexer simile a quello visto in ingresso al PLL), sia i registri visti per il PLL per generare in uscita la frequenza richiesta. Per consultare lo schema integrale per le sorgenti di clock fare riferimento al datasheet citato in precedenza (pag. 94/847).

5. Real Time Clock

Il Real Time Clock (RTC) è un orologio binario che fornisce la data e ora corrente. In questo caso utilizza come sorgente di clock l'LSI RC che fornisce una frequenza di 32kHz che viene in seguito, tramite il settaggio di prescaler, ridotta a 1Hz. All'avvio del microcontrollore la data e l'ora vengono settate automaticamente a: "17:30:00 MER 29-12-20", questo perché l'RTC non possiede una fonte elettrica che possa tenerlo in funzione anche quando il microcontrollore non viene alimentato. La data e ora possono essere facilmente cambiate tramite seriale accedendo al menu Data/Ora come visto nel paragrafo 3 "Interfaccia". Il software è in grado di leggere solo determinati formati.

I formati ammessi sono i seguenti:

Data: GG-MM-AA

Ora: HH:MM:SS

Giorno: basta che siano scritte le prime due lettere del giorno (es. MA per martedì) sia in maiuscolo che in minuscolo.

6. Termistore

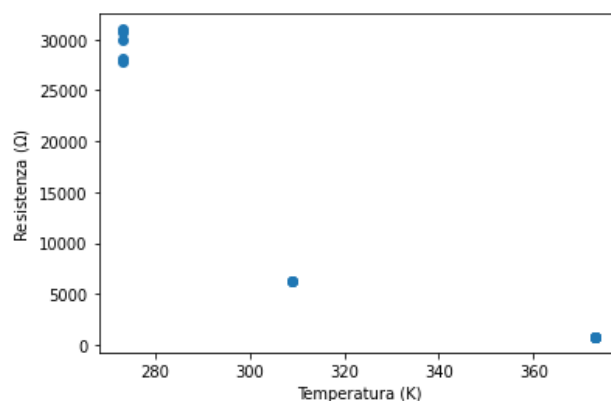
Come già anticipato nel Paragrafo 1, la temperatura ambiente non viene letta tramite il sensore di temperatura integrato all'interno della scheda NUCLEO-F401RE, ma bensì grazie ad un termistore esterno.

Come prima cosa è stato caratterizzato il termistore. Per fare ciò sono stati misurati, tramite un multimetro, i valori della resistenza del termistore a tre temperature diverse:

1. 100° C
2. 0° C
3. 35.9°

Si è sfruttato il punto di ebollizione dell'acqua, che avviene a 100° C a pressione atmosferica, e il punto di congelamento dell'acqua, che avviene 0° C a pressione atmosferica. Per i 35.9° C invece è bastato misurare la temperatura corporea. Di seguito il grafico delle misurazioni effettuate:

Come si vede dal grafico delle misurazioni (fig. 9), il termistore è un termistore di tipo NTC.



(Fig. 9 Grafico misurazioni effettuate termistore)

Per il calcolo della temperatura è stata utilizzata l'equazione di Steinhart-Hart:

$$\frac{1}{T} = A + B \ln(R) + C [\ln(R)]^3$$

Bisognava calcolare i coefficienti A, B e C del suddetto termistore. Per fare ciò è stata sfruttata una libreria Python

(https://scikitlearn.org/stable/auto_examples/linear_model/plot_bayesian_ridge.html) che utilizza la regressione.

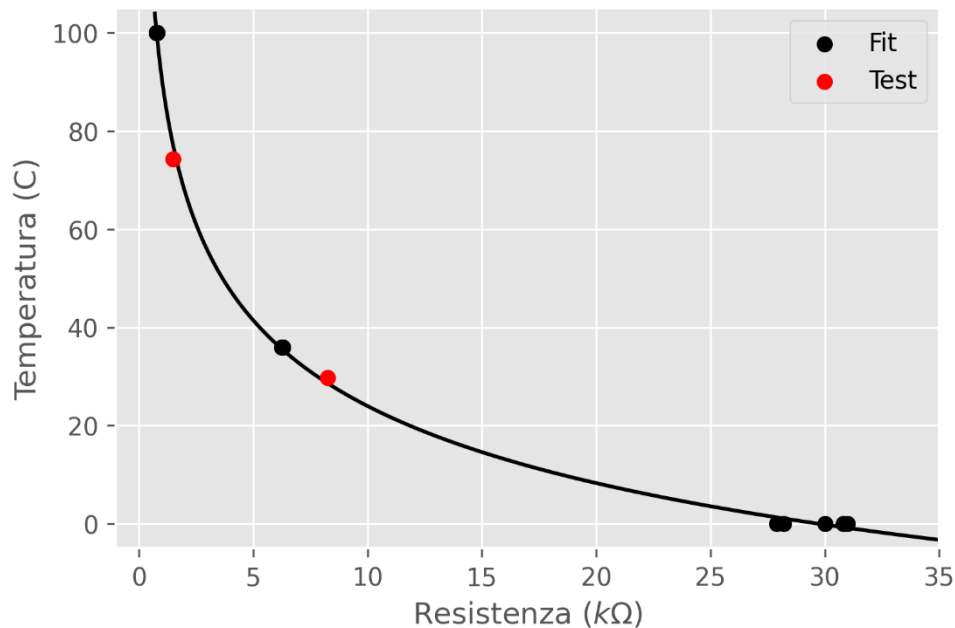
I coefficienti calcolati sono:

A: 9.20425684e-04

B: 2.63029364e-04

C: 2.89615702e-08

Utilizzando i coefficienti appena trovati è stata calcolata la curva del termistore e testata con due misurazioni indipendenti:

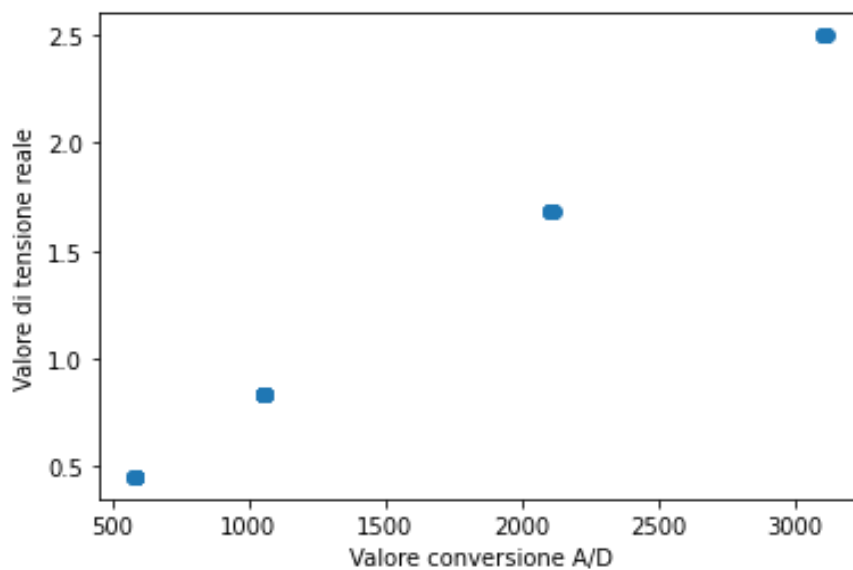


(Fig. 10 Curva caratteristica del termistore)

In seguito, per leggere il valore di resistenza del termistore, è stato collegato il termistore in serie ad una resistenza da 10kΩ a cui è stata applicata una tensione di 5V. Il valore di tensione che cade sul termistore viene letto dalla porta I/O del microcontrollore, a cui è stato applicato una conversione A/D. Il valore risultante dalla conversione non rappresentava il reale valore di tensione misurato, ma vi è una relazione lineare tra i due. Utilizzando lo stesso metodo per calcolare la curva caratteristica del termistore, abbiamo sostituito al termistore resistenze di valore noto (nello specifico 10kΩ, 5.1kΩ, 2kΩ, 1kΩ) e letto i valori restituiti dalla conversione A/D. L'equazione risultante è la seguente:

$$y = 0.00081167x - 0.02647693$$

dove y rappresenta la tensione reale che cade sul termistore e x è il valore restituito dalla conversione A/D.



(Fig. 11 Retta relazione tra valore conversione A/D e tensione su termistore)

Una volta trovata la tensione che cade sul termistore, applicando la formula inversa del partitore di tensione si calcola la R del termistore:

$$R = \frac{10k\Omega * V_0}{5 - V_0}$$

dove V_0 rappresenta il valore di tensione che cade sul termistore

Una volta trovata la R, basta sostituirla nell'equazione di Steinhart-Hart per calcolare il valore della temperatura corrispondente (in °K).

Il valore della temperatura viene monitorato ogni secondo dal software e nel caso in il valore registrato superi i 30° C viene fatto scattare un allarme (fare riferimento al Paragrafo 10).

7. LCD 16x02

Alla scheda NUCLEO-F401RE è stato collegato uno schermo LCD 1602a. Questo è un display a due linee con 16 caratteri per ognuna. Lo schermo presenta in totale 14 pin descritti nella seguente immagine:

Pin 1-2: Alimentazione

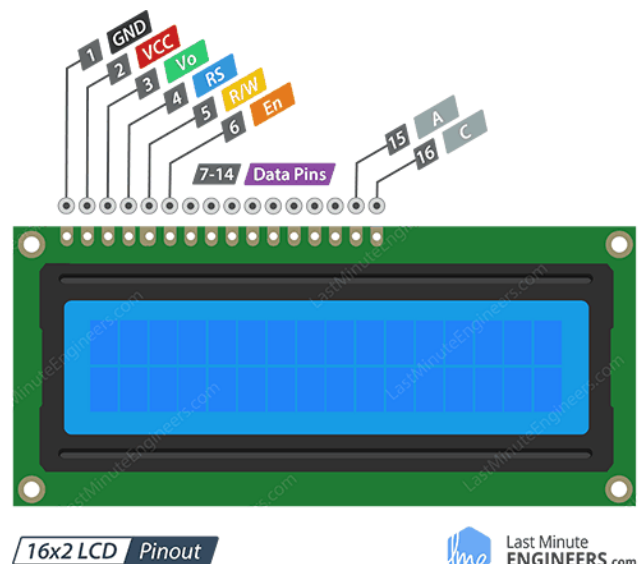
Pin 3: Contrasto

Pin 4-5: Selezione tipo di Operazione

Pin 6: Abilita lettura/scrittura dei dati

Pin 7-14: 8 bit data

Pin 15-16: Retroilluminazione



Last Minute
ENGINEERS.com

(Fig. 12 Schema Pin LCD)

Questo display LCD può essere settato sia per utilizzare una sola linea che entrambe, inoltre può ricevere i dati utilizzando tutti gli 8 bit data, ma anche in modalità 4bit che permette di utilizzare solo gli ultimi 4 bit per il trasferimento delle informazioni.

Nel caso di questo progetto, come si evince dalla Fig. 1, il display è stato collegato per funzionare in modalità 4bit, inoltre è possibile controllare il contrasto e la retroilluminazione tramite i due potenziometri, rispettivamente potenziometro A e potenziometro B.

Tutte le istruzioni per settare il display su 2 linee a 16 caratteri ed in modalità 4bit vengono svolte in automatico dal software all'avvio.

In questo progetto il display viene utilizzato per stampare o la data e l'ora corrente o la temperatura. Per quanto riguarda la prima stampa, sulla prima linea del display viene stampata l'ora corrente in formato HH:MM:SS, mentre sulla seconda viene stampata la data nel formato GGG GG:MM:AA. Per quanto riguarda invece la stampa della temperatura, sulla prima linea viene stampata la temperatura nel seguente formato "TEMPERATURA:XX C", mentre la seconda viene riempita in base alla temperatura misurata. Se la temperatura è compresa tra i 15 e i 30 gradi, per ogni grado sarà riempito un carattere sul display. Se verrà registrata una temperatura crescente sarà visualizzata sul display una sorta di barra a riempimento. Una volta raggiunto il limite superiore, corrispondente a 30° C, scatterà un allarme e contemporaneamente sul display sarà visualizzato un messaggio di allarme.

NOTA: Il valore della temperatura sul display viene aggiornato ogni tre secondi

NOTA2: Il valore della temperatura viene costantemente filtrato con un filtro IIR passa-basso a singolo polo

8. Tasto USER (BLU)

Il tasto USER è stato programmato per far sì che ogni volta che venga premuto il display mostri informazioni differenti. Nello specifico, sul display può essere stampata o la data e l'ora corrente oppure la temperatura attuale. Quindi ogni qual volta il tasto viene premuto si passa dalla stampa della data a quella della temperatura e viceversa.

9. LED

La scheda NUCLEO-F401RE ha integrato un led, indicato sulla stessa, come LED2. L'utente tramite il Menu Led, affrontato nel Paragrafo 3, può scegliere diverse modalità di funzionamento per lo stesso.

NOTA: Se durante l'utilizzo della funzione di lampeggio viene richiesto di cambiare la frequenza del processore (vedi Paragrafo 4), il led sarà spento e sarà necessario riattivare la funzione di lampeggio (vedi Paragrafo 3), nel caso l'utente lo desideri.

10. Active Buzzer

Alla scheda NUCLEO-F401RE è stato collegato un Active Buzzer. Questo svolge la funzione di sistema di allarme nel caso in cui la temperatura superi il valore preimpostato di 30° C. Infatti, nel caso in cui quest' ultima eventualità occorra viene inviato al buzzer un segnale di clock con periodo uguale a un secondo fin quando la temperatura non ritorna in valori normali, cioè al di sotto dei 30° C.

11. Criticità dell'attuale implementazione

Durante la fase di test del software implementato sono state riscontrate delle criticità che sono state gestite ma non risolte del tutto. Queste criticità riguardano due particolare ambiti.

Comunicazione seriale

Può succedere che la comunicazione seriale non vada a buon fine e che il microcontrollore non riesca a riconoscere la scelta effettuata dall'utente. All'interno del software è stato implementato un metodo per riconoscere quando questa eventualità si verifichi e di conseguenza verrà chiesto all'utente di ripetere la risposta fin quando quest'ultima non sarà riconosciuta come una risposta valida.

Gestione degli interrupt

Dal momento che l'intera comunicazione tra periferiche e microcontrollore è stata gestita tramite interrupt (stampa su lcd, monitoraggio temperatura, attivazione buzzer, utilizzo del pulsante USER, risposta utente su seriale) potrebbe verificarsi che più interrupt possano trovarsi in stato di pending contemporaneamente. Dal momento che il nostro processore può servire un interrupt per volta, potrebbe succedere che una di queste venga eseguita in un tempo diverso da quello previsto. Non è stata prevista una differenziazione delle priorità dei suddetti interrupt in quanto si è pensato che questa potrebbe portare al verificarsi di un ulteriore problema: gli interrupt con una priorità inferiore potrebbero non essere mai eseguite o eseguite molto raramente.

Per spiegare meglio questo concetto riferiamoci direttamente al software implementato in questo progetto. Possiamo identificare due insiemi di interrupt:

1. Sistema di allarme (Monitoraggio temperatura + attivazione buzzer)
2. Interazione con l'utente (Stampa Data/Ora + Stampa Temperatura + Pulsante USER + Seriale)

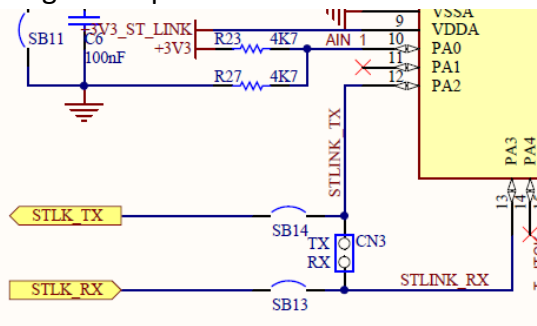
Verrebbe naturale pensare di dare una priorità maggiore al sistema di allarme.

Facciamo il caso che ci troviamo in una situazione di allarme e che sul display sia visualizzata la temperatura. Abbiamo quindi l'interrupt del monitoraggio della temperatura che si verifica ogni secondo, l'interrupt del buzzer che si verifica ogni mezzo secondo e l'interrupt della stampa che si ripete ogni tre secondi. Avere una differenza di priorità tra le due tipologie di interrupt significa, secondo il ragionamento espresso in precedenza, che l'interrupt che stampa il valore della temperatura sul display resti in stato di pending per molto più tempo di quanto sia il programmatore che l'utente si aspetti.

Per questi motivi, non riuscendo a trovare nuove soluzioni nell'immediato, si è pensato di lasciare tutti gli interrupt con priorità uguale.

12. Future Implementazioni

1. Rendere il collegamento seriale più stabile utilizzando dell'hardware dedicato per poter implementare il controllo tramite i pin RTS e CTS. Questo perché come è possibile vedere dallo schematic della scheda e dal datasheet "stm32f401re" i suddetti pin non sono collegati alla porta USB dell' ST-LINK.



(Fig. 13 Collegamento STLINK)

PA0	I/O	FT	(5)	USART2_CTS, TIM2_CH1/TIM2_ETR, TIM5_CH1, EVENTOUT
PA1	I/O	FT	-	USART2_RTS, TIM2_CH2, TIM5_CH2, EVENTOUT
PA2	I/O	FT	-	USART2_TX, TIM2_CH3, TIM5_CH3, TIM9_CH1, EVENTOUT
PA3	I/O	FT	-	USART2_RX, TIM2_CH4, TIM5_CH4, TIM9_CH2, EVENTOUT

(Fig. 14 Descrizione Pin USART)

2. Progettare un sistema di alimentazione esterno per l'RTC
3. Migliorare l'efficienza energetica e implementare l'attivazione e lo spegnimento della "sleep mode" in base all'interazione con l'utente
4. Implementare una funzione che permette di cambiare la data e l'ora utilizzando il pulsante USER
5. Implementare una funzione che aggiorni la data e l'ora ogni qual volta che il microcontrollore viene connesso ad un altro dispositivo (es. PC), nel caso in cui ce ne fosse bisogno
6. Utilizzare la modalità "Input Capture Mode" dei TIM per leggere la reale frequenza di sistema del microprocessore
7. Utilizzare il buzzer per far "suonare" al microcontrollore delle canzoni
8. Migliorare la sincronizzazione delle varie Interrupt che gestiscono le varie componentistiche hardware collegate alla scheda.
9. Aumentare la frequenza massima selezionabile dall'utente gestendo, però, i prescaler in modo dinamico da software per permettere a tutte le periferiche di funzionare in modo corretto. (Ad esempio, l'USART2 ed alcuni TIM supportano una frequenza massima di 42MHz mentre l'USART1 ed altre periferiche supportano una frequenza di clock massima di 84MHz. Fare riferimento al datasheet "RM0368_stm32f401xbc-and-stm32f401xde".

Alternativa di progetto: sostituire l'active buzzer con un passive buzzer e gestirlo tramite la funzione PWM dei TIMx