Sommario

Utilizzare schermo TFT per Rasperry con Arduino 3
Introduzione 3
Preparazione della libreria 3
Corrispondenza PIN e segnali 3

Utilizzare schermo TFT per Rasperry con Arduino

Introduzione

Da tempo volevo riutilizzare un vecchio schermo LCd Touch specifico per Raspberry che non usavo più, nello specifico si tratta di un schermo da 3.5 pollici con risoluzione 480×320 gestito da un drive ILI9486 senza nessun riferimento, vedi immagine seguente



L'idea era di utilizzarlo come schermo per progetti con Arduino o meglio ancora con una scheda WiFi come la WeMos D1 mini che oltre ad essere molto economica è veramente piccola.

Il problema stava nel trovare una libreria (in quelle standard dell'IDE non ci sono) per poterlo pilotare. Cercando sul Web mi sono imbattuto in un'ottima libreria la TFT_eSPI che oltre a gestire una miriade di drive sia per Arduino che per ESP8266, ma anche per molte altre piattaforme, ha dei profili ottimizzati proprio per questo schermo. Purtroppo è priva di documentazione ma ha un ricco corredo di esempi da cui capirne il funzionamento.

Preparazione della libreria

Questa libreria, potendo lavorare con moltissimi schermi e almeno 5 piattaforme, ha bisogno di essere personalizzata per l'uso che se ne deve fare ovvero:

- piattaforma ⇒ ESP8266
- schermo ⇒ RPi LCD Touch 9486

La personalizzazione consiste nel modificare il file “User_Setup_Select.h” presente nella root della libreria dove si dovrà decommentare l'istruzione “Include” relativa allo “User_Setup” che vogliamo usare, attenzione che ci deve essere un solo include attivo, quindi si dovranno commentare eventuali altri include presenti.

Di file “User_Setup” la libreria ne ha già 40 preimpostati ma, nell'eventualità, c'è anche un template per crearne uno personale. Scorrendo le varie possibilità ho visto che una poteva fare al caso mio, ero veramente fiducioso in quanto nella cartella “Tools” è presente una immagine del mio LCD con la pinnatura per il collegamento, purtroppo non era così facile, niente da fare non ne voleva sapere di funzionare, faceva qualcosa ma non di sensato.

Senza perdermi d'animo ho iniziato a studiare i vari file della libreria e mi sono accorto che tra la WeMos e la ESP8266 ci sono piccole differenze nell'assegnazione dei GPIO, ho quindi creato un mio User_Setup con l'assegnazione esplicita dei GPIO mancanti e con grane soddisfazione ha iniziato a visualizzare il contenuto dello schermo.

Corrispondenza PIN e segnali

PIN LCD	Segnale	BUS	PIN WeMos
1	3.3V		
2	5V		5V
3	NC		

PIN LCD	Segnale	BUS	PIN WeMos
4	5V		
5	NC		
6	GND		
7	NC		
8	NC		
9	GND		
10	NC		
11	TP_IRQ		
12	NC		
13	NC		
14	GND		
15	NC		
16	NC		
17	3.3V		
18	LCD_RS	DC	D3
19	LCD_SI/TP_SI	MOSI	D7
20	GND		
21	TP_SO	MISO	D6
22	RST	RST	D4
23	LCD_SCK/TP_SCK	SCK	D5
24	LCD_CS	CS	D8
25	GND		G
26	TP_CS	T_CS	D2

Fonte:
Wiki - Progetti

Autore:
Walter62

Ultimo aggiornamento: **2023/01/01 17:03**