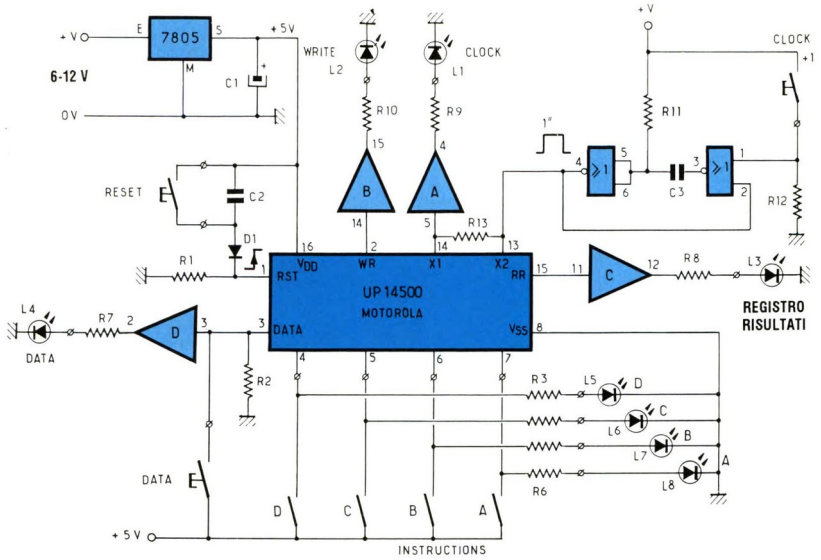
PROGETTO DIDATTICO CON PROCESSORE MC14500

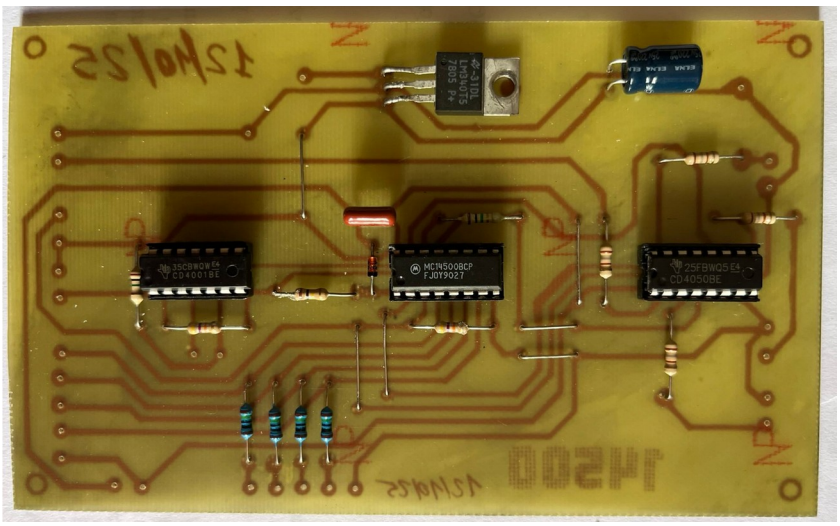
Da INTERNET:
Il Motorola MC14500B è un'unità di controllo industriale (ICU - Industrial Control Unit) a 1 bit, introdotta nel 1976. A differenza dei microprocessori tradizionali, l'MC14500B è progettato per applicazioni di logica booleana sequenziale, operando su singoli bit di dati.

La mia idea deriva dopo aver visto il progetto sulla rivista Fare Elettonica n° 87 del 1992.
Io sono sempre stato affascinato dal linguaggio macchina e in questo progetto si può testare questo mondo.

Schema elettronico presentato dalla rivista.

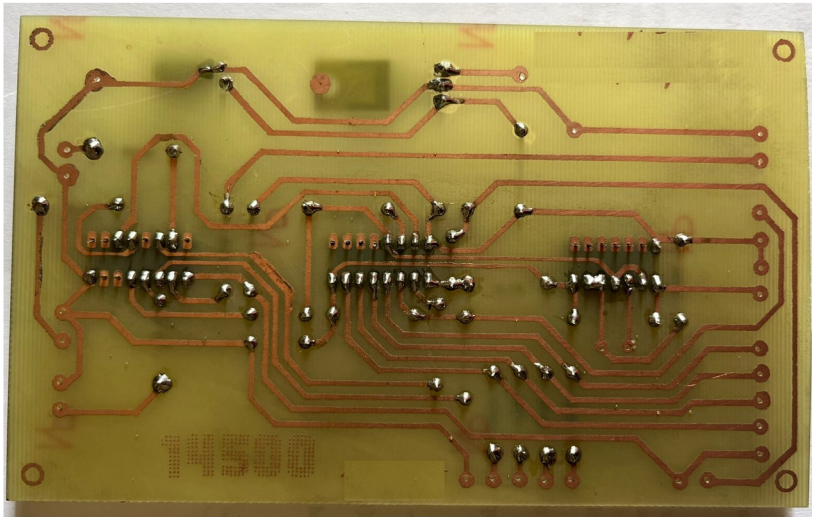


Di seguito è riportata la descrizione del PCB da me realizzato:
A sinistra integrato IC4001 con porte NOR per evitare impulsi spuri del pulsante di CLOCK, al centro il processore MC14500, a destra IC4050 buffer per controllo dei led. Nel mio progetto ho aggiunto una basetta con un altro IC4050 per vedere, trami led, lo stato delle Flag del processore. Oltre a resistenze e condensatori c'è uno stabilizzatore di tensione a 5Volt.

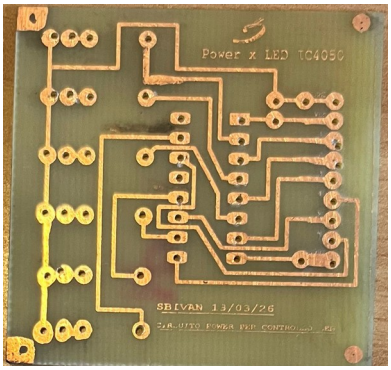


I

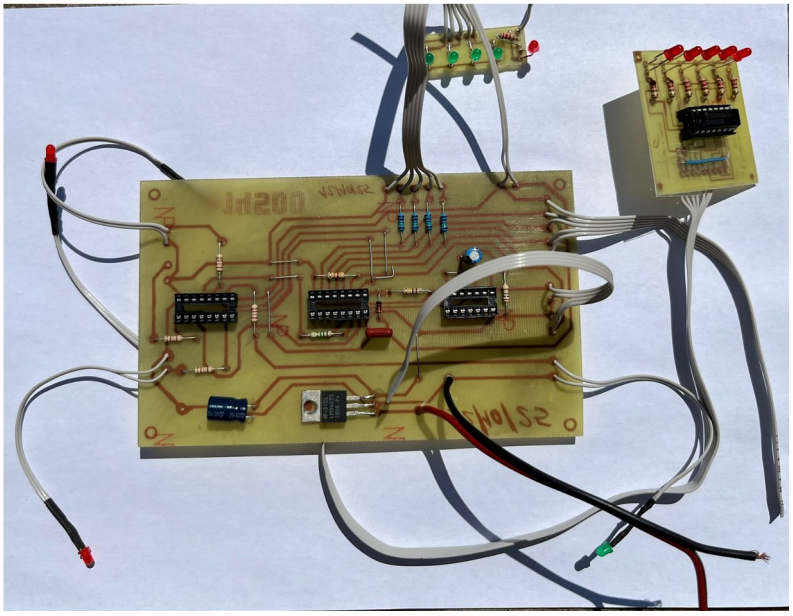
PCB lato saldature.



Basetta per monitorare lo stato delle Flag dell’MC14500 con IC4050.



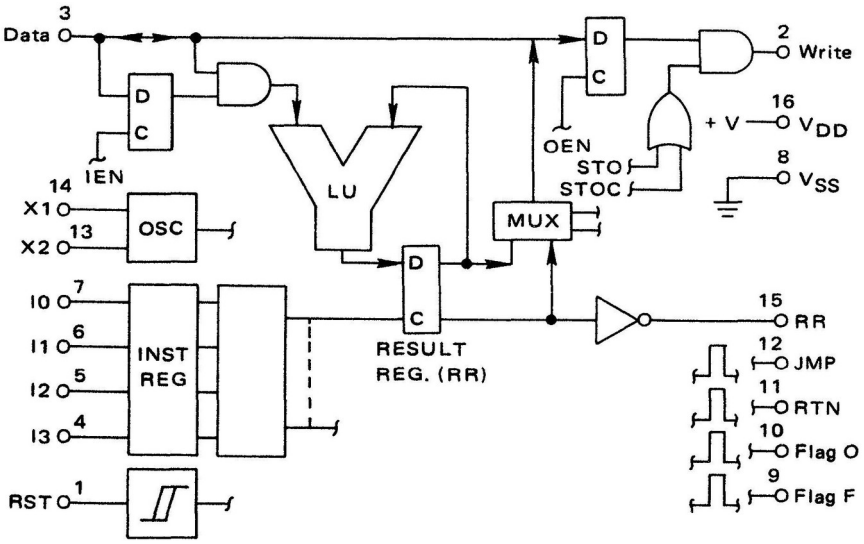
Progetto in fase di assemblaggio, la basetta a destra buffer IC4050 con 5 led rossi, come già detto, per vedere lo stato delle Flag e ulteriore basetta per avere i 5 led uniti per evitare troppi fili volanti.



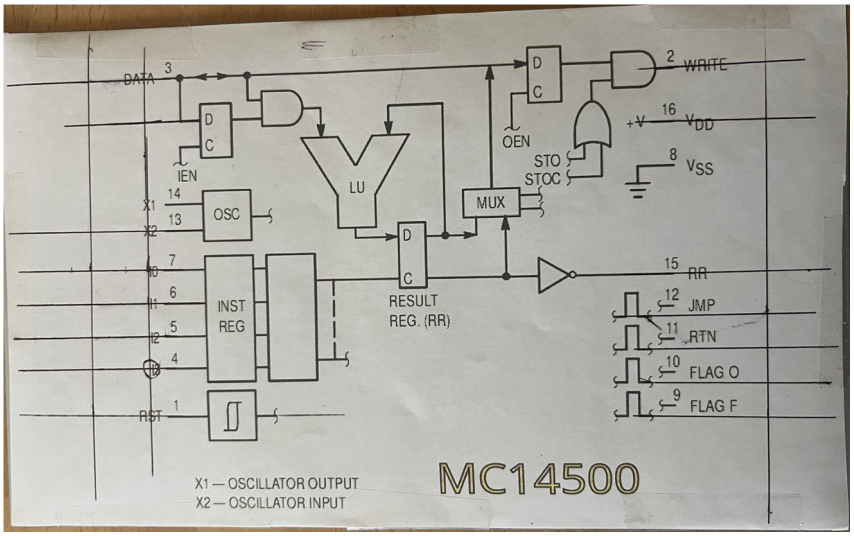
Per completare il progetto ho utilizzato un alimentatore switching da 12 V, 1 A, sfruttando lo stabilizzatore 7805 già presente sulla scheda.



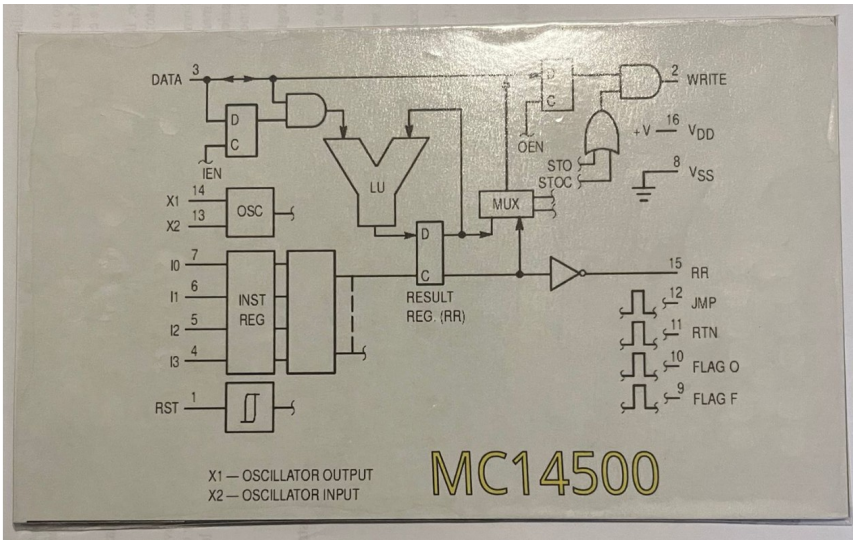
Serigrafia usata per la mascherina.



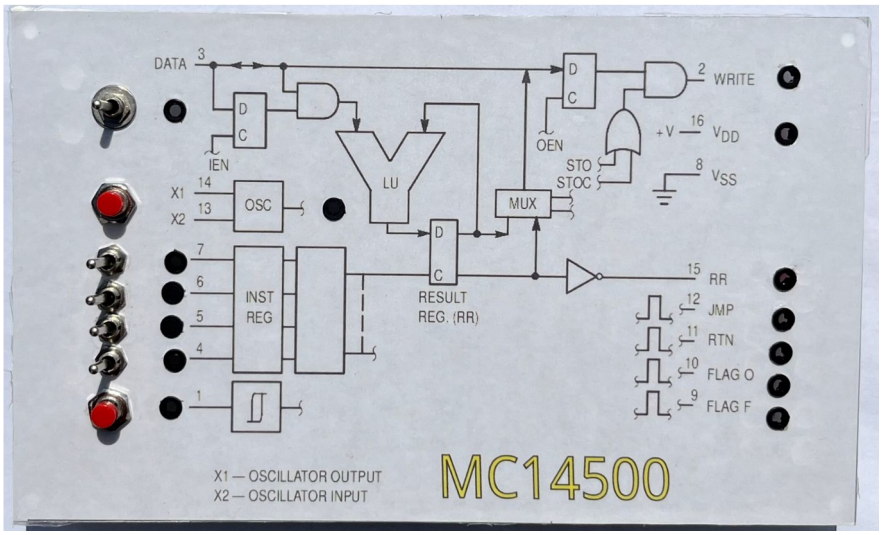
Mascherina usata per segnare i fori.



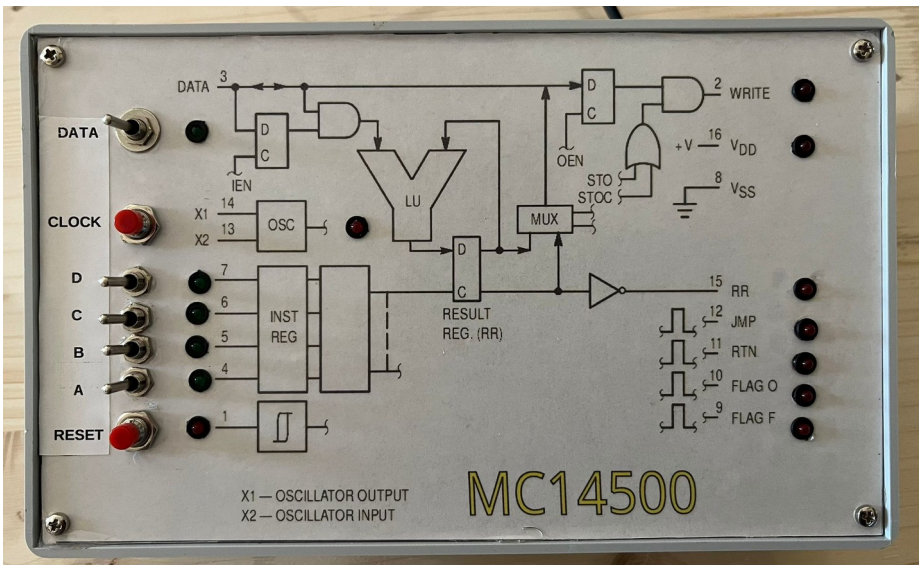
Definitiva con fori sulla mascherina di alluminio(sotto la carta).



Mascherina con: interruttori, pulsanti e led inseriti.



Progetto finito.



Istruzioni del microprocessore MC14500

Table 1. MC14500B Instruction Set

Instruction Code		Mnemonic	Action
0	0000	NOPO	No change in registers. RR → RR, Flag O → ⎍
1	0001	LD	Load result register. Data → RR
2	0010	LDC	Load complement. $\overline{\text{Data}}$ → RR
3	0011	AND	Logical AND. RR • Data → RR
4	0100	ANDC	Logical AND complement. RR • $\overline{\text{Data}}$ → RR
5	0101	OR	Logical OR. RR + Data → RR
6	0110	ORC	Logical OR complement. RR + $\overline{\text{Data}}$ → RR
7	0111	XNOR	Exclusive NOR. If RR = Data, RR → 1
8	1000	STO	Store. RR → Data Pin, Write → ⎍
9	1001	STOC	Store complement. $\overline{\text{RR}}$ → Data Pin, Write → ⎍
A	1010	IEN	Input enable. Data → IEN Register
B	1011	OEN	Output enable. Data → OEN Register
C	1100	JMP	Jump. JMP Flag → ⎍
D	1101	RTN	Return. RTN Flag → ⎍ and skip next instruction
E	1110	SKZ	Skip next instruction if RR = 0
F	1111	NOPF	No change in registers. RR → RR, Flag F → ⎍

Esempi programmi dalla rivista Fare Elettronica:

a) Funzione AND a 3 ingressi

1010	IEN	convalida dell'ingresso
1011	OEN	convalida dell'uscita
0001	LD	caricare DATA 1
0011	AND	funzione AND con DATA 2 e DATA 1, inserire risultato in RR
0011	AND	funzione AND con DATA 3 e contenuto di RR, caricar risultato in RR
1000	STO	caricare il risultato all'uscita e convalidare WRITE.

b) Funzione NAND a 2 ingressi

1010	IEN	
1011	OEN	
0001	LD	caricare DATA 1
0011	AND	funzione AND con DATA 2 e DATA 1, caricare risultato in RR
1000	STOC	caricare il complemento del risultato all'us cita e convall dare WRITE

c) Funzione EXOR a 2 ingressi

1010	IEN	
1011	OEN	
0001	LD	
0111	XNOR	funzione EXNOR tra DATA 2 e DATA 1
1001	STOC	caricare il complemento del risultato

d) Funzione NOT

1010	IEN	
1011	OEN	
0001	LD	
1001	STOC	

e) Funzione memoria (comando marcia/ arresto)

1010	IEN	
1011	OEN	
0001	LD	caricare DATA 1 (stato del pulsante: on)
0101	OR	realizzare una funzione OR con RR e disporla in RR
0100	ANDC	realizzare una funzion e AND con i, complemento di DATA 2 (stato del pulsante : off)
1000	STO	caricare il risultato all'uscita (stato della memoria.