

ELETTRONICA DEI SISTEMI DIGITALI

LABORATORIO # 6 “MICROCONTROLLORI: COMUNICAZIONI SERIALI”	a.a. 2006-2007	DATA: 16/19 Gennaio 2007
--	----------------	--------------------------

TAVOLO:	SQUADRA (elenco nomi): 1 2 3 4 5
----------------	--

INFORMAZIONI SUI COMPONENTI UTILIZZATI:

PDIP

(RESET) PC6	1	28	PC5 (ADC5/SCL)
(RXD) PD0	2	27	PC4 (ADC4/SDA)
(TXD) PD1	3	26	PC3 (ADC3)
(INT0) PD2	4	25	PC2 (ADC2)
(INT1) PD3	5	24	PC1 (ADC1)
(XCK/T0) PD4	6	23	PC0 (ADC0)
VCC	7	22	GND
GND	8	21	AREF
(XTAL1/TOSC1) PB6	9	20	AVCC
(XTAL2/TOSC2) PB7	10	19	PB5 (SCK)
(T1) PD5	11	18	PB4 (MISO)
(AIN0) PD6	12	17	PB3 (MOSI/OC2)
(AIN1) PD7	13	16	PB2 (SS/OC1B)
(ICP1) PB0	14	15	PB1 (OC1A)

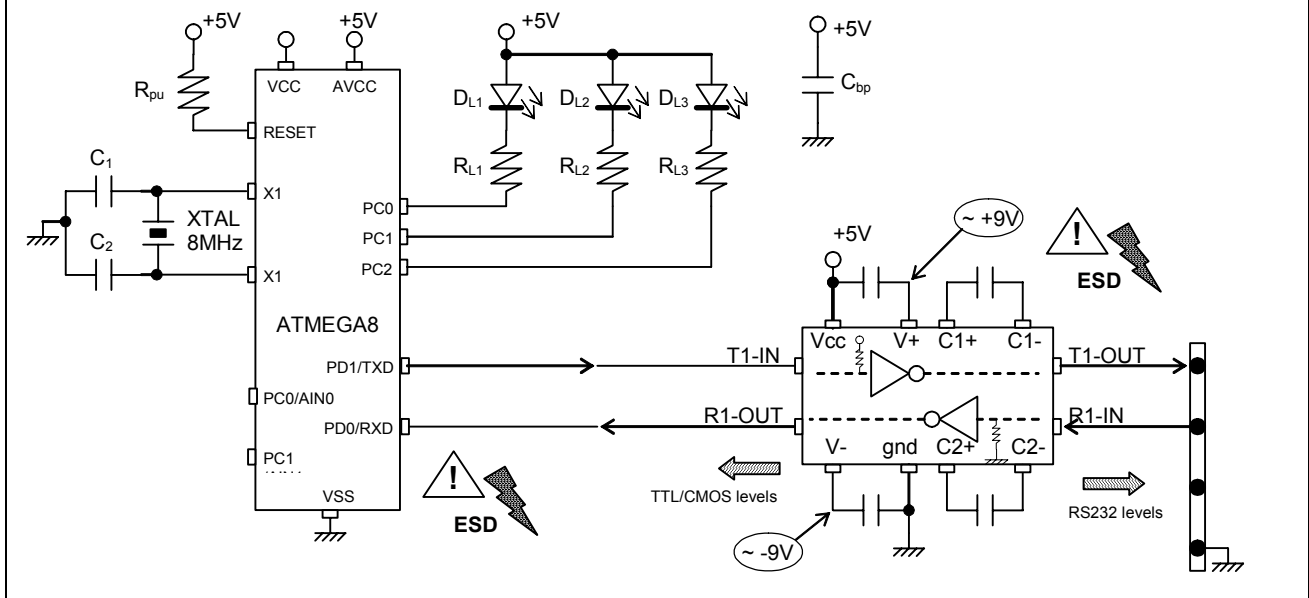
ATMEL ATMEGA8

ICL232 / MAX232

ESD

1. Ricordate di mantenere e poi restituire **integro** il microcontrollore... le masse sempre collegate per prima cosa, attenzione anche a quella del **cavetto seriale!** Verificate che l'alimentatore abbia una sufficiente **limitazione di corrente** per fare funzionare il vostro circuito, in questo modo ridurrete al minimo le probabilità di danni irreversibili. Non superate i valori di tensioni ammessi per il dispositivo, alimentate con 5V. Attenzione che alcune versioni del chip **MAX232 o equivalente soffre l'ESD !!**

2. **Montate il circuito base con tre led** da pilotare (rosso, giallo e verde), ed il circuito di interfaccia TTL/CMOS <=> RS232: verificate tensioni (anche sul MAX232 come da specifiche su *datasheet*), corrente assorbita, frequenza oscillatore, stato dei segnali al *power-on*. Oltre al cavetto per la programmazione su LPT1, collegate dopo la verifica e programmazione il cavetto seriale alla porta 'COM1' del PC. Per la comunicazione con il microcontrollore utilizzate il programma di terminale remoto 'terminal' con parametri di comunicazione: COM1: 9600,8,N,1, no-echo, no-control, ANSI.



3. Analizzate il programma fornito, che in funzione di un carattere ASCII di comando impartito da PC, accende il LED rosso (R) o il LED giallo (G) o il LED verde (V).
Il programma opera a 9600 baud, 8 bit, nessuna parità, un bit di stop. Il codice esegue *echo* dei caratteri inviati, e ritorna messaggi differenti nei vari casi, un messaggio di *sign-on* di accensione ed un messaggio di errore se il carattere non risulta tra quelli ammessi.

4. *PutChar*: Modificate il codice per realizzare una uscita seriale continua di un carattere a scelta, analizzate con l'oscilloscopio il formato del segnale seriale (start, dati e stop).
Verificate la ricezione corretta dei dati sul PC con il programma di gestione terminale (HyperTerminal).

5. *GetChar*: Modificate il codice per realizzare un programma che riceva caratteri ASCII dal PC, e faccia, con la subroutine *GetChar*, *echo* verso il PC (modalità *remote loop-back*).

6. *PrintString*: Scrivete un programma semplice che invii un messaggio a vostra scelta al PC, usando come carattere terminatore il NULL (00 hex).