

La comunicazione seriale

Quando due sistemi a microprocessori devono comunicare fra loro ad una distanza superiore ad un paio di metri, la comunicazione parallela non è affidabile poiché vi sono accoppiamenti capacitivi tra i singoli fili conduttori che tendono a cortocircuitare le varie linee fra loro.

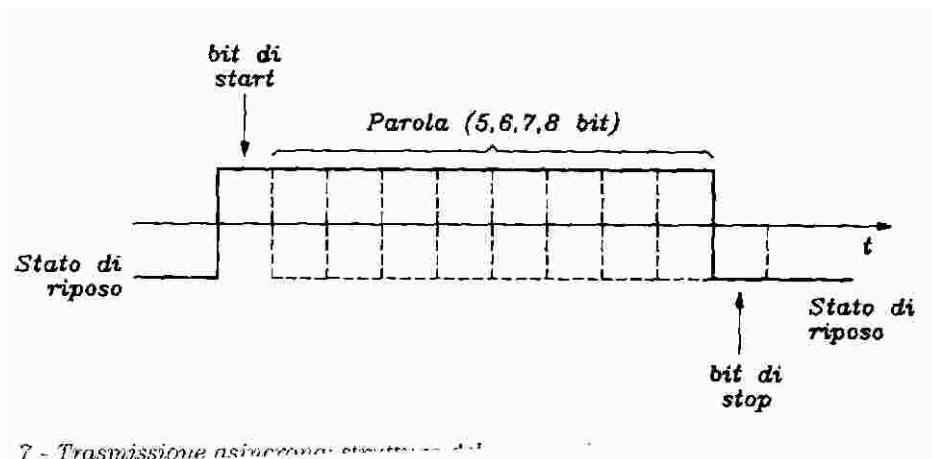
poiché gli accoppiamenti fra le varie linee sono capacitivi e le reattanze capacitive tendono a diminuire con la frequenza dei segnali, i problemi manifestati da una comunicazione parallela aumentano all'aumentare della distanza di collegamento e della frequenza dei segnali trasmessi.

Per tali ragioni, per collegamenti a lunga distanza, viene adottata la comunicazione seriale nella quale i dati vengono trasmessi bit a bit su una sola linea. Anche in questo caso, se la distanza da coprire è superiore ai 30/40 metri, la qualità della trasmissione scade per cui occorre fare riferimento alle tecniche di modulazione.

Si hanno due tipi di trasmissione seriale: Trasmissione sincrona e asincrona

Nella trasmissione asincrona i due sistemi che stanno dialogando fra loro usano due clock diversi per cui non si può garantire che i due clock siano in fase o siano alla stessa frequenza.

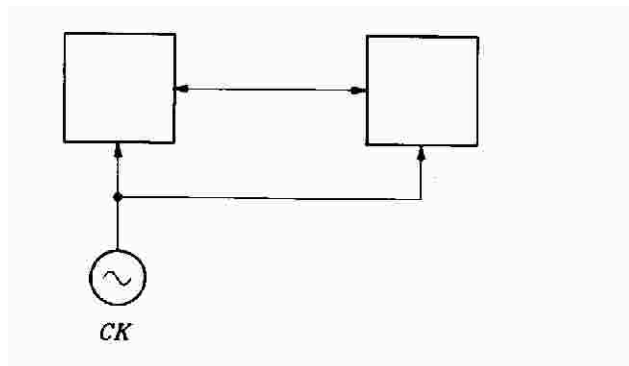
IL ricevitore non ha alcuna informazione sulla temporizzazione con cui arrivano i dati. Per garantire che trasmettitore e ricevitore siano sincronizzati, le informazioni da inviare sono suddivise in piccoli blocchi di 5, 6, 7 o 8 bit che sono precedute e seguite da bit di sincronizzazione detti bit di start e di stop



E' chiaro che l'aggiunta di questi bit di controllo diminuisce l'efficienza della trasmissione poiché vi è un elevato rapporto fra bit di controllo e bit che contengono informazioni. Questo tipo di trasmissione si usa allora quando i dati vanno inviati in maniera sporadica. Ogni blocco è costituito allora da

- Un bit di start che individua il passaggio dalla condizione di riposo alla condizione di trasmissione
- Il blocco dei dati che possono essere da 5 a 8 più un eventuale bit di parità
- 1, 1,5 o 2 bit di stop che indicano la fine della trasmissione.

Nella trasmissione sincrona trasmettitore e ricevitore utilizzano per la trasmissione e la ricezione lo stesso clock. Tale clock viaggia su una linea apposita (vedi circuiti C113 e C114 dell'interfaccia RS 232) oppure è estratto mediante appositi circuiti del modem dai dati ricevuti (vedi circuito C115).



```
int led = 5;

char val = '0';

void setup() {

  pinMode(led,OUTPUT);

  Serial.begin(9600);

}

void loop () {

  val = Serial.read();

  if(val=='a') {

    digitalWrite(led,HIGH);

    Serial.println("led ON");

  }

  else if(val=='b') {

    digitalWrite(led,LOW);

    Serial.println("led OFF");

  }

}
```

```
int incomingByte = 0; // for incoming serial data

void setup() {

  Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

void loop() {

  // send data only when you receive data:

  if (Serial.available() > 0) {

    // read the incoming byte:

    incomingByte = Serial.read();

    // say what you got:

    Serial.print("I received: ");

    Serial.println(incomingByte, DEC);

  }

}
```

```

int i,j,v[4]; // for incoming serial data

int numero;

void setup() {

    Serial.begin(9600); // opens serial port, sets data rate to 9600 bps

    Serial.println("inserisci il numero di 4 cifre ");

    j=0;

    for(i=0;i<4;i++) v[i]=0;
}

void loop() {

    // send data only when you receive data:

    if (Serial.available() > 0) {

        // read the incoming byte:

        v[j] = Serial.read()-48;

        // say what you got:

        Serial.print("I received: ");

        Serial.println(v[j], DEC);

        j++;

    } // say what you got:

    numero =1000*v[0]+100*v[1]+10*v[2]+1*v[3];

    if (j==4){

        Serial.print("ho ricevuto: ");

        Serial.println(numero);

        Serial.println("inserisci il numero di 4 cifre");

        numero=0;j=0;

        for(i=0;i<4;i++) v[i]=0;

    }

}

```