Azzolini Riccardo 2020-12-04

Interfacce seriali asincrone

1 Interfacce digitali

Le **interfacce digitali** permettono il trasferimento di parole digitali tra due o più dispositivi, che possono essere microcontrollori, sensori, computer, ecc. Esistono due principali tipi di interfacce:

- le interfacce **parallele**, che fanno uso di una linea (un filo) per ogni bit della parola (più alcune linee di controllo);
- le interfacce **seriali**, nelle quali i bit di una parola vengono inviati uno dopo l'altro su poche linee.

Nell'ambito dei microcontrollori, le interfacce parallele non sono ormai quasi più usate, perché l'elevato numero di linee necessarie porta a esaurire in fretta i piedini di I/O del microcontrollore, rendendo difficile il collegamento di molti dispositivi. Si preferiscono dunque le interfacce seriali, che possono a loro volta essere suddivise in due categorie:

- interfacce seriali **sincrone**, nelle quali il trasferimento dei dati viene sincronizzato mediante una linea di *clock*, permettendo a uno dei dispositivi collegati (il *master*) di determinare la velocità di trasferimento;
- interfacce seriali **asincrone**, nelle quali non si ha un clock per la sincronizzazione, quindi i dispositivi collegati devono conoscere a priori la velocità di trasferimento oppure "mettersi d'accordo" in qualche modo.

Tipicamente, i microcontrollori usano le interfacce asincrone per comunicare con dispositivi come i computer, e le interfacce sincrone per comunicare con altri componenti digitali più semplici, come ad esempio i sensori a uscita digitale.

Un'altra distinzione è quella tra le interfacce seriali half-duplex e full-duplex:

- **half-duplex** significa che la comunicazione può avvenire in un solo verso alla volta, e per questo può bastare anche solo un filo;
- **full-duplex** significa che la comunicazione può avvenire contemporaneamente in entrambi i versi, il che richiede almeno due fili (RX, per la ricezione, e TX, per la trasmissione).

2 RS-232

RS-232 è un'interfaccia seriale asincrona full-duplex, che in passato era molto diffusa sui computer, ma adesso è stata in gran parte sostituita da USB. Essa è un'interfaccia *point-to-point*, cioè permette il collegamento di due soli dispositivi.

Tipicamente, i dati vengono inviati in parole di una lunghezza che può variare tra i 5 e i 9 bit. Ciascuna parola è preceduta da un bit di start, un bit di parità opzionale (per il rilevamento degli errori), e infine uno o due bit di stop. Anche la velocità di trasmissione, che misurata in *baud* (bit al secondo), è configurabile. Perché due dispositivi possano comunicare correttamente, devono avere entrambi la stessa identica configurazione (baud, numero di bit di dati, presenza del bit di parità e numero di bit di stop).

3 RS-485

RS-485 è un'interfaccia seriale asincrona usata prevalentemente in campo industriale. Essa può funzionare in modalità half-duplex con due fili, o in modalità full-duplex con quattro fili, ed è RS-485 un'interfaccia *multipoint* o *multidrop*: agli stessi fili possono essere connessi anche più di due dispositivi.

Rispetto a RS-232, RS-485 supporta velocità più alte e cavi più lunghi (fino a 1200 m, mentre RS-232 è limitata a circa 15 m).

4 USB

L'interfaccia seriale asincrona **USB** (*Universal Serial Bus*) è quella ormai più usata per interfacciare i computer a dispositivi esterni. Essa permette la connessione di più dispositivi a una sola porta, mediante l'uso di appositi hub, e può anche fornire ai dispositivi collegati una corrente di alimentazione.

5 Comunicazione seriale asincrona con Arduino

Ogni scheda Arduino ha una o più coppie di piedini RX e TX che fungono da interfacce seriali asincrone configurabili (*UART*, *Universal Asynchronous Receiver Transmitter*). Anche la comunicazione con il computer a cui Arduino è collegato avviene mediante un'interfaccia seriale; poiché i computer moderni non hanno porte seriali, questa viene emulata tramite USB (si parla di VSP, Virtual Serial Port, o VCP, Virtual COM Port):

- sul computer, l'emulazione viene eseguita da un apposito driver;

- su Arduino, l'emulazione può essere svolta dal microcontrollore stesso (nel caso dei microcontrollori più avanzati, che supportano direttamente USB) oppure da un dispositivo dedicato, montato sulla scheda e connesso a una delle coppie di piedini che fungono da interfaccia seriale (tipicamente i piedini digitali 0 e 1).

L'IDE di Arduino fornisce un "monitor seriale" che permette di interagire con Arduino tramite quest'interfaccia.

Nel codice Arduino, per inizializzare la porta seriale si usa il comando `Serial.begin`, che riceve come argomenti le opzioni di configurazione della porta:

- la velocità, in baud (ad esempio 9600);
- opzionalmente, una costante che specifica il numero di bit di dati, la presenza del bit di parità (che può essere pari o dispari) e il numero di bit di stop; il valore di default è `SERIAL_8N1`, che significa 8 bit di dati, nessun bit di parità, e 1 bit di stop.

La configurazione specificata deve essere identica a quella dell'altro dispositivo comunicante.

Dopo aver inizializzato la porta, si possono usare numerosi comandi per leggere e scrivere dati (`Serial.read`, `Serial.print`, ecc.[1]).

Infine, la porta seriale può essere disattivata con `Serial.end`. Questo permette, ad esempio, di usare i piedini corrispondenti alla porta come normali piedini di I/O.

[1] L'elenco completo dei comandi per la comunicazione seriale è disponibile alla pagina https://www.arduino.cc/reference/en/language/functions/communication/serial/