

Esperienza n°1: confronto logica cablata (integrata) – logica programmabile

L'esperienza ha come scopo il confronto fra la logica cablata (integrata) e la logica programmabile. Un led viene acceso mediante la pressione di un bottone e il tempo per cui questo rimane acceso è definito da:

- logica cablata: valore della resistenza e della capacità del condensatore inseriti nel circuito
- logica programmabile: valore della variabile impostata nel programma

L'esperienza mostra come sia più semplice modificare tale valore mediante la logica programmabile.

Descrizione:

- Scaricare mediante l'IDE Arduino il programma TimedButton.pde
- Alimentare la breadbord collegandola opportunamente ad Arduino
- Collegare opportunamente i cavi bottone e led ad Arduino (attenzione ai pin da utilizzare, verificare nel programma)
- Modificare i tempi di accensione del LED:
- logica programmabile: modificare valore variabile
- logica cablata: modificare il valore di resistenza equivalente mettendo la resistenza fornita in serie o in parallelo alla resistenza già presente nel circuito e modificare il valore di capacità equivalente mettendo il condensatore fornito in serie o in parallelo al condensatore già presente nel circuito.

Esperienza n°2: utilizzo di Arduino come controllore per la temperatura raggiunta da una cella Peltier

Lo scopo dell'esperienza è il monitoraggio della temperatura raggiunta da una cella Peltier, misurata mediante un trasduttore lineare a contatto con il lato caldo della cella. Nel programma sono impostabili due valori di soglia: warning e allarme. I valori delle temperature acquisite sono visualizzate a schermo.

Se la temperatura misurata è inferiore alla soglia di warning, Arduino accende il led verde e il programma visualizza a schermo la temperatura in verde. Se la temperatura è compresa fra la soglia di warning e quella di allarme, Arduino accende il led giallo e il programma visualizza a schermo la temperatura in giallo. Se la temperatura è superiore alla temperatura di allarme, Arduino accende il led rosso ed attiva un allarme sonoro, il programma visualizza a schermo la temperatura in rosso.

Descrizione:

- Alimentare la breadbord
- Aprire nella IDE Arduino il programma: File/Examples/Firmdata/Standard Firmdata e scaricarlo su Arduino

- Aprire in Processing il programma ControlloTemperatura/Graph2D_adv/Graph2D_adv.pde e lanciarlo mediante il tasto Play
- Alimentare la cella Peltier (ATTENZIONE: non superare 3 V e 1 A, altrimenti la cella si rompe)

Esperienza n°3: misure di accelerazione su mensola vibrante

Con l'esperienza proposta si vuole mostrare il funzionamento di due importanti aspetti della comunicazione e gestione dei segnali di trasduttori di misura in un sistema industriale:

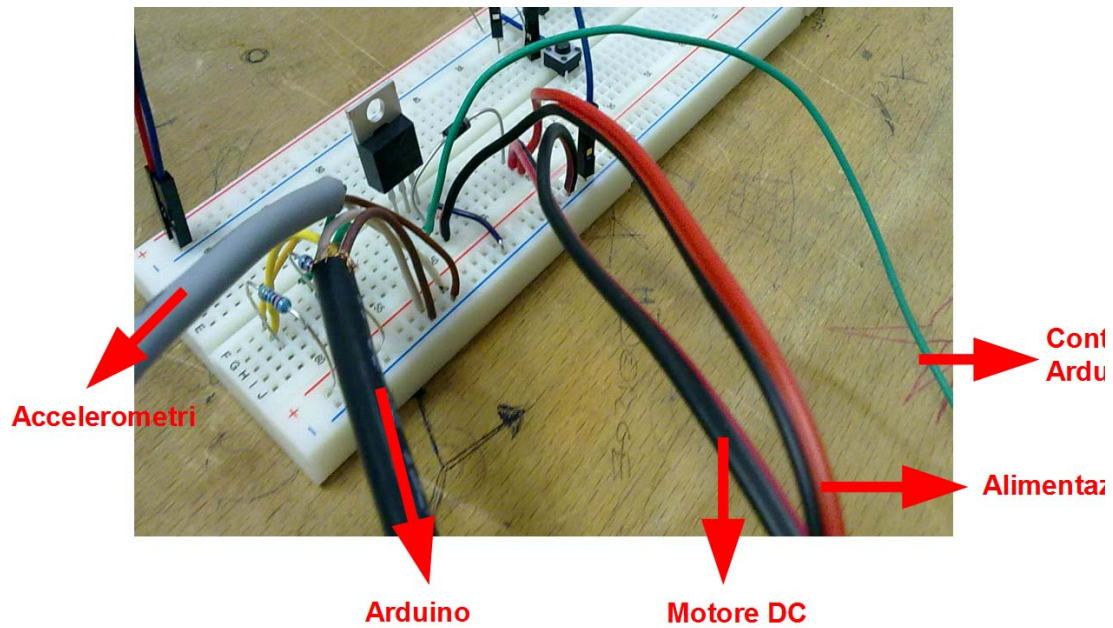
- la comunicazione dei dati tramite bus di campo, in particolare utilizzando il protocollo I2C;
- scambio di dati tramite porta seriale RS232, standard diffuso praticamente in ogni tipo di dispositivo di misura, controllo o sistema di monitoraggio.

Nel dettaglio, si propone di acquisire le vibrazioni di una mensola in acciaio tramite due accelerometri (modello BMA180) utilizzando il bus di campo I²C.

La piattaforma Arduino interroga una alla volta i due sensori, tramite i relativi indirizzi, acquisisce i due valori già digitalizzati dai sensori e li comunica al PC attraverso la porta seriale attraverso un convertitore USB-Seriale già integrato sulla scheda. Dall'altro lato, un programma PC sviluppato con Processing riceve in ingresso i valori numerici delle accelerazioni e li visualizza a monitor sia in forma di grafico sia come valori numerici.

La trave è posta in vibrazione da un motore a corrente continua con una massa eccentrica. La velocità di rotazione del motore è controllata tramite il software Processing. E' infatti presente una scrollbar con cui impostare questo parametro tra un minimo (0%) ed un massimo (100%). Questo valore è inviato dal PC al controllore Arduino sempre tramite porta seriale, realizzando così una comunicazione bidirezionale, ed attuata tramite una modulazione di tipo PWM.

ATTENZIONE ALLE OSCILLAZIONI DEL SISTEMA VIBRANTE !



Per avviare procedere come segue:

- collegare il cavo degli accelerometri (cavo quadripolare con guaina nera) secondo lo schema:
- bianco: +3.3V;
- marrone: GND;
- giallo: SDA (Analog pin 4);
- verde: SDK (Analog pin 5);

Come sono collegate le resistenze di pullup?

- Collegare il cavo di controllo del motore (filo verde singolo) al Digital pin 9;
- Alimentare il motore DC:
- filo rosso: +24V;
- filo nero: 0V;
- avviare l'IDE di Arduino e caricare, selezionando la porta seriale corretta, il

programma “Accelerazione_I2C_BMA180_versione03”, nella memoria del microcontrollore;

- aprire l'IDE di Processing e caricare il programma “EsercitazioneMotoreArduino”. Controllare che nella funzione `void setup()` la variabile `String PortName = "COMX"` sia settata con lo stesso nome di quella dell'IDE di Arduino;
- avviare il software premendo “Play”;
- impostare la velocità di rotazione del motore tramite la slidebar ed osservare i valori numerici delle accelerazioni (in milli-g) sul grafico. Quale dei due segnali proviene dall'accelerometro più lontano dall'incastro?