10 Novembre 2023

**4 Lezione Embedded**

Sistema microcontrollore -> economico -> tanti pulsanti devo mettere tanti pulsanti nel microcontrollore

Ingressi -> punti dell’integrato (piedini) -> ogni posto più c’è più costa -> diversi metodi -> con 7 fil, 12 segmenti, rileva 20 pulsanti diversi

Pulsante -> sistema a matrice -> con 4 righe -> dipende pulsante da rilevare, incroci che voglio generare -> metto pulsante che genera un cortocircuito tra riga nessima e bottone mesimo

Pulsante non collegato -> determinata tensione -> PULL DOWN -> resistenza collegata a 5V -> tensione stabile e sicura -> sistema che commuta in base allo stato del bottone H / L (alto o basso)

Decido io valori -> input e output

Qualsiasi pressione -> non comporta nessun cambiamento -> pulsante collegato a 5V -> non c’è corrente che passa -> non rilevo nessuna modifica parte D -> stato di riposo tastierino

Con tutto 1 -> non sta accendo nulla

Problema -> 3 punti

Portando a zero prima dei punti -> garantendo altre linee a 1 -> cortocircuitando diversi valori. Il dato che porto a 0 lo collego a massa e quindi a 0 -> controllare diversi dati una alla volta e posso decidere il suo valore di quello che sto decidendo di leggere

Quando premo pulsante diventa -> 0 -> controllo se bit è andato a zero -> loop annidato che controlla se bi è andato a zero e tengo traccia -> indice che mi serve per tenere in memoria i dati dove mi trovo

Un ciclo con 12 controlli -> tutti bit 3 per RD0 con i relativi RB0, RB1 e RB2 cosi per RD1 e RD2 -> controllo per tutte le colonne

Codice corrispondente alla funzione che capisce i controlli delle intersezioni dove è a 0

For (RB0 RB1 RB2)

{

For(RD)

{

If(RD == 0)

{

Nriga + ncol \* 4; // combinazione lineare spazio di due

coordinate su due dimensioni,

numero univoco ciascun pulsante

}

}

}

1 for -> abbassare 1 bit alla volta -> decido ultimi 3 bit in base a cosa leggo

2 for -> leggo la colonna del bit 3 finali corrispondenti secondo quindi leggo il secondo

PORTB & = ~ (1 << col);

<< -> shift logico -> byte (8 bit) -> colonne tutte a zero tranne quello che mi interessa per creare il 1

Shift a sx -> spostarlo a sx -> in questo modo va avanti -> numero dopo -> va via -> numero dopo diventa a 0 -> dipende macchina di solito 0

Numero di volte che mi sposto shift -> sposto numero di quanto mi serve

~ -> 0 diventa 1

1 diventa 0

& = te stesso e quello che viene dopo e lo aggiungi

PORTB = PORTB & ~ (1 << col);

io posso decidere -> cosa forzare a 1, abbasso una colonna alla volta -> controllo bit -> dopo controllo valore della resistenza del valore -> ~ inverto valore

PULL UP -> premo per accendere, READ -> lettura per accendere

Ultimo for -> controllo ciascun bit delle righe -> 1 shiftato a sx e in questo modo trovo RD0, RD1 oppure gli altri -> infine calcoli pulsante indicato