3 Novembre 2023

**3 Lezione Embedded**

Timer -> avanza partendo frequenza attivazione macchina che va al quarzo e incrementa i valori dei dati -> perscaler -> divide i valori e li mette a posto, frequenza ridotto sincronizzata -> contatore conta con il tm0 -> con 256 cancella esegue interrupt -> fa partire funzione di interrupt

Intrupt cadenza regolare -> tempi prestabiliti -> non c’entra quanto codice scrivo

Interrupt -> va avanti indifferente dal codice

Interrupt -> condizioni che non posso modificare -> posso lavorare in modo da decidere quante metterne in overflow (con 255) -> posso decidere interrupt con tempi di scatti -> tarare tempi che ci interessano in modo da ridurre gli interrupt

Tempo tra interrupt e quello dopo più precisi -> velocità limite, funzionamento della macchina

Timer 1 -> più lungo difficilmente utilizzatati

In base a quanto causare interupt -> devi gestire

TMRO =

PS -> più grande possibile affinché TMRO sia positivo

Scheda Pig Genius -> microcontrollore 877

877 -> meglio di 628 più porte fino alla porta E

PortA -> 5 bit

Portb -> 8 bit

PortC -> 8 bit

PortD -> 8 bit

PortE -> 3 bit

Puoi programmare 32 bit in totale

Ci sono anche 2 Relè -> collegati a RC0 (1) e RE0 (2)

1 ventola -> illumina dati, più si illumina più ventola va veloce

Resistore che serve per andare più veloce

Leggere temperatura Lm35 -> leggi temperatura alzi o scaldi

Balzer -> dispositivo che fa suono

Display -> scrivi frasi -> due righe 16 bit caratteri alfanumerici tabella ascii

16 \* 2 oppure 16 \* 4

2 righe o 4 righe

Ho anche due manopole -> regolo tensione che posso regolare, mantengo in memoria dell’ora

RTC -> conta tempo

EEPROM -> memoria tenere all’accensione

KeyPad -> tecnica rilevamento dato (schermo touch) idea uguale

Display -> matrie righe e colonne, ineriscano tramite incroci, file delle matrici collegate ai 5V

Negli incroci ci sono i pulsanti, se non premo tasti e ho fili collegati in pullup, fili come input -> si legge sempre 1 -> caduta di tensione è 0V

PortB -> impostata output -> impongo io H / L

In PortB -> se etto valori alti sono sempre accessi -> posso capire la pressione da applicare

Ci sono bit più e meno significativi