

## [main del pic16F88]

```
Void main (void) {

    init_OSC();
    init_PORT();
    init_ADC();
//    init_LAMA();
    I2cInterrupt();
    I2cInitSlave(ADDR_PIC);
    attesaSc(1)
    Fai_Misure();

    Stato=99;
    while(Stato==99) continue;          // attende un comando I2C

// può uscire solo in stato 0 o 1
    while (1){
        switch (Stato) {
            case 0:
{ VERDEBLI;TMR1ON=1;attesaSc(7);VH1=248;VL1=20;Stato=1;}
            case 1:{// sequenza principale
                    Fai_Misure();      }      break;
            case 2:{StopTEMP();} break;
            case 3:{StopBATT();} break;
            case 4:{StopTutto();} break;
            default :{InERR();}break;
        }
    }
}
```

L'errore avviene se attivo la linea init\_LAMA() che attiva il PWM passando per il ramo [case 0:].

Ti allego la routine Fai\_Misure() di seguito che genera l'errore:

```
void Fai_Misure () {
    NMOTO=0;
    TMAX=Misura_PIN(0); MIS_OUT[1]=TMAX;
    TEMP=Misura_PIN(1);
    if (TEMP<TMAX) {TMAX=TEMP; NMOTO=1;}
    TEMP=Misura_PIN(2);
    if (TEMP<TMAX) {TMAX=TEMP; NMOTO=2;}
    TEMP=Misura_PIN(3);
    if (TEMP<TMAX) {TMAX=TEMP; NMOTO=3;}
    TEMP=Misura_PIN(4);
    if (TEMP<TMAX) {TMAX=TEMP; NMOTO=4;}
    VBATT=Misura_PIN(5);
    // VERIFICA ERRORE con LED - senza lama funziona.
    if (VBATT>=0 && VBATT<=10) {ROSSOBLI;} else GIALLOBLI;
    MIS_OUT[0]=VBATT;
    MIS_OUT[2]=TMAX;
    MIS_OUT[3]=NMOTO;
}
```

La routine ISR è la seguente:

```
void interrupt isr(void) {
if (SSPIF && SSPIE) {
    // COM = xx(DA) (P) - (S) (RW)x(BF)
    // COM = xx 0 x - 1 0 x 1 =9 --> !DA & S & !RW & BF indir scrive (9)
    // COM = xx 1 x - 1 0 x 1 =41 --> DA & S & !RW & BF dato scrive (41)
    // COM = xx 0 x - 1 1 x 0 =12 --> !DA & S & RW & !BF Indir per leggere (12)
    // COM = xx 1 x - 1 1 x 0 =44 --> DA & S & RW & !BF dato legge (44)
    // COM = xx 1 x - 0 0 x 0 =32 --> DA & !S & !RW & !BF (nack) (32)
```

```

COM=SSPSTAT & 0B00101101;
if (COM==9) { // Arriva un INDIR da Master (!DA) al fine di scrivere (!RW) dati su Slave
    SSPBUF=SSPBUF; // LETTURA per AZZERARE BF
    if(SSPOV) {SSPOV = 0;} // Gestione Overflow
}
else if (COM==41) { // Arriva un DATO da Master (DA) per essere scritto su Slave
    DATOIN=SSPBUF;// mette in DATOIN e AZZERA BF
    switch (DATOIN) {
        case 0: {Stato=0;} break; // ciclo avvio lama
        case 1: {Stato=1;} break; // ciclo principale
        case 2:{ Stato=2;} break; // Temp ALTA
        case 3:{ Stato=3;} break; // BATT BASSA
        case 4: {Stato=4; }break; // Ferma Tutto
        default : {
            PARAM_IN[ind_in++]=DATOIN;
            ind_in=ind_in % 2;
            Stato=1;
        }
    }
    if(SSPOV) { SSPOV = 0; ROSSOON;} // Gestione Overflow
} else if(COM==12) { // INDIR da Master (!DA) al fine di leggere (RW) dati da
Slave
    SSPBUF=MIS_OUT[ind_out++]; // preleva da DATOIN e AZZERA BF
    ind_out=ind_out % 4;
    CKP=1; // Rilascia il clock
} else if(COM==44) { // da Master una richiesta di lettura (RW)
di un DAT (DA)
    SSPBUF=MIS_OUT[ind_out++]; // preleva da
DATOIN e AZZERA BF
    ind_out=ind_out % 4;
    CKP=1; // rilascia il clock
} else if (COM==32) {
    // Fine comunicazioni NACK da Master
}

SSPIF=0;
}

// Timer generatori del PWM a 50 HZ
if (TMR0IE && TMR0IF) {
    LAMA=1;
    TMR0=V;
    TMR1H=VH1;
    TMR1L=VL1;
    TMR0IF=0;
}
if (TMR1IE && TMR1IF) {
    LAMA=0;
    TMR1H=VH1;
    TMR1L=VL1;
    TMR1IF=0;
}
}

```