

SITO DI RIFERIMENTO PER COMPILARE ED ESEGUIRE IL CODICE:

ALCUNI dei possibili progetti di domani (i gruppi saranno creati in maniera casuale):

- 1) Montaggio e controllo del robot umanoide "Robotis Mini". Se possibile scaricate e installate su un pc portatile il software per farlo muovere (in caso di necessità il docente vi darà un pc).
- 2) Pilotare il rotore di coda di un elicottero attraverso un transistor NPN
- 3) Controllare la tensione di un circuito elettrico tramite regolazione del duty cycle di un segnale in base ad un transistor NPN
- 4) Controllare il movimento di un servomotore in base alla luce rilevata con un fotoresistore
- 5) Progetto a sorpresa

GRUPPI:

- 1) Gordiani, Milleti, Venditti, Fiumi, Venditti (Fermi), Jin, Salvatori, Montanaro, Castelli, Sayad
- 2) Sauli, Boschi, Papale, Rossi, Gregorio, Di Carmine, Santacchini, Brian
- 3) Borelli, Silvani, Elena L., Frasca, Martellucci, Cacciamani, Lo Bartolo, Venanzi, Albanesi
- 4) Di Legge, Scervo, Ciullo, Sabatini, Iannetti, Salle, Carla ., Petrucci
- 5) Cammou, Moreschi, D'Andrea, De Vitis, Giorgio ., Chiriaco, Vincenti

IDE on-line: <https://www.tinkercad.com/>

(Simili: <https://scratch.mit.edu/projects/editor/?tutorial=getStarted>, <https://www.mblock.cc/en-us/>)

SLIDES INTRODUTTIVE:

- [ArduinoPartePrima](#)
- [ArduinoParteSeconda](#)

ESERCIZI:

1)

//int piedino = 13; //ho definito una variabile intera e inizializzata con il valore 13

#define piedino 13 //ho definito il parametro di nome piedino a cui associo il valore 13

void setup()

{

pinMode(piedino, OUTPUT); //definisco il piedino n.13 come uscita digitale

}

void loop()

{

digitalWrite(piedino, HIGH);

delay(5); // Wait for 1000 millisecond(s)

digitalWrite(piedino, LOW);

delay(5); // Wait for 1000 millisecond(s)

}

2) IF:

```
#define piedino 13 //ho definito il parametro di nome piedino a cui associo il valore 13
```

```
int pippo = 0; //definisco una variabile intera di nome pippo
```

```
void setup()
```

```
{  
  pinMode(piedino, OUTPUT); //definisco il piedino n.13 come uscita digitale  
  pippo = 10;  
}
```

```
void loop()
```

```
{  
  if(pippo >= 10)  
  {  
    digitalWrite(piedino, HIGH);  
    delay(1000); // Wait for 1000 millisecond(s)  
    digitalWrite(piedino, LOW);  
    delay(1000); // Wait for 1000 millisecond(s)  
  }  
}
```

3) IF con condizioni logiche

```
#define piedino 13 //ho definito il parametro di nome piedino a cui associo il valore 13
```

```
int pippo = 0; //definisco una variabile intera di nome pippo
```

```
void setup()
```

```
{  
  pinMode(piedino, OUTPUT); //definisco il piedino n.13 come uscita digitale  
}
```

```
void loop()
```

```
{  
  if( (pippo >= 2) && (pippo <= 7) )  
  {  
    digitalWrite(piedino, HIGH);  
    delay(1000); // Wait for 1000 millisecond(s)  
  }  
}
```

```
digitalWrite(piedino, LOW);
delay(1000); // Wait for 1000 millisecond(s)
}

pippo = pippo + 1;

}
```

4) Comunicazione seriale + IF

#define piedino 13 //ho definito il parametro di nome piedino a cui associo il valore 13

int pippo = 0; //definisco una variabile intera di nome pippo

```
void setup()
{
  pinMode(piedino, OUTPUT); //definisco il piedino n.13 come uscita digitale
  Serial.begin(9600); //inizializzo la porta seriale alla velocità di 9600 "parole" al secondo
}
```

```
void loop()
{
  if( (pippo >= 2) && (pippo <= 7) )
  {
    Serial.println("Sono nell IF");
    digitalWrite(piedino, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(piedino, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
  }
}
```

pippo = pippo + 1;

Serial.println(pippo);

}

5) operazione modulo e OR

#define piedino 13 //ho definito il parametro di nome piedino a cui associo il valore 13

```

int pippo = 0; //definisco una variabile intera di nome pippo

int resto = 0; //variabile intera

void setup()
{
  pinMode(piedino, OUTPUT); //definisco il piedino n.13 come uscita digitale
  Serial.begin(9600); //inizializzo la porta seriale alla velocità di 9600 "parole" al secondo
}

void loop()
{
  //accendo e spengo il led solo se p è pari oppure uguale a 731
  resto = pippo % 2;
  Serial.println(resto);

  if( (resto == 0) || (pippo == 731))
  {
    Serial.println("Sono nell IF");
    digitalWrite(piedino, HIGH);
    delay(1000); // Wait for 1000 millisecond(s)
    digitalWrite(piedino, LOW);
    delay(1000); // Wait for 1000 millisecond(s)
  }

  pippo = pippo + 1;

  Serial.println(pippo);

  Serial.println("Fine funzione loop");
  delay(1000);
}

```

6) Visibilità delle variabili

```

int b = 0; //varibile con visibilità globale

void setup()
{

```

```

//così definita la variabile a è visibile solo all'interno della
//funzione setup() [visibilità locale]
int a = 0; //variabile di tipo intero definita della funzione setup
Serial.begin(9600); //inizializzo la porta seriale alla velocità di 9600 "parole" al secondo
}

```

```

void loop()
{
  //Serial.println(a); //errore di visibilità della variabile
  Serial.println(b);
  delay(1000);
}

```

7) Ancora visibilità delle variabili

```
int b = 0; //varibile con visibilità globale
```

```

void setup()
{
  //così definita la variabile a è visibile solo all'interno della
  //funzione setup() [visibilità locale]
  int a = 0; //variabile di tipo intero definita della funzione setup
  Serial.begin(9600); //inizializzo la porta seriale alla velocità di 9600 "parole" al secondo
  b = 1; //b è visibile globalmente
}

```

```

void loop()
{
  int a = 3; // attenzione questa variabile a è diversa da quella
  //definita nella setup(), sono indipendenti e visibili
  //ognuna nella propria funzione.

  //Serial.println(a); //errore di visibilità della variabile
  Serial.println(b+a);
  delay(1000);
}

```

8) TIPI DI DATO- cast

```

float a = 0;
int b = 3;

float c = 0.0;

void setup()
{

    a = 2;

    Serial.begin(9600);
}

void loop()
{

    //b = a/3;
    //c = a*0.4;
    c = ((float) b) *0.4; //conversione (cast) a float del contenuto della  variabile b prima
dell'operazione
    Serial.println(c);

}

```

9) Cast con operazioni iterative

```

int b = 1;
float c = 0.0;

void setup()
{

    Serial.begin(9600);
}

void loop()
{

    c = c + ((float) b) / 2.0; //conversione (cast) a float del contenuto della  variabile b prima
dell'operazione
    Serial.println(c);
    delay(1000);
}

```

```
}
```

10) //somma dei numeri da 0 a 100

```
int a = 0;
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
    for(int k=0;k<=100;k++)
```

```
    {
```

```
        a = a + k;
```

```
        Serial.println(a);
```

```
        delay(50);
```

```
    }
```

```
}
```

11) dente di sega di altezza 300 in 2 s (periodo di 4s)

```
int k = 0;
```

```
void setup()
```

```
{
```

```
    Serial.begin(9600);
```

```
}
```

```
void loop()
```

```
{
```

```
    Serial.println(k);
```

```
    delay(20/3); //la funzione delay non prende numeri in virgola mobile
```

```
    //ho bisogno di attendere meno di un millisecondo quindi
```

```
    //devo usare delayMicroseconds()....
```



```

    if(k >= 300)
    {
        k = 0;
    }
    k++;

}

```

12) Sinusoide di periodo 1 secondo e ampiezza 5. Esercizio...

13) FILTRAGGIO dei segnali

```

float time = 0.0;
#define omega1 2.0*3.14/1.0 //pulsazione del segnale vero
#define omega2 2.0*3.14/0.01 //pulsazione angolare del rumore additivo
#define a1 5 //ampiezza del segnale che voglio misurare
#define a2 0.3 // ampiezza del rumore additivo
float s = 0.0;

// FILTRO
#define theta 0.8
float f = 0.0;

void setup()
{
    Serial.begin(9600);
}

void loop()
{
    //segnale misurato (prodotto sintenticamente ora nel programma)
    s = a1*sin(omega1*time) + a2*sin(omega2*time);

    //uso del filtro del primo ordine
    f = theta*f + (1-theta)*s;

    Serial.println(f); //stampo il valore del filtro
}

```

```
    delay(1);  
    time = time + 0.001; //incremento il tempo  
}
```

14) Scelta condizionata dalla misura (filtrata)

```
float time = 0.0;  
#define omega1 2.0*3.14/1.0 //pulsazione del segnale vero  
#define omega2 2.0*3.14/0.01 //pulsazione angolare del rumore additivo  
#define a1 5 //ampiezza del segnale che voglio misurare  
#define a2 0.5 // ampiezza del rumore additivo  
float s = 0.0;
```

```
#define pinLed 13
```

```
// FILTRO  
#define theta 0.8  
float f = 0.0;
```

```
void setup()  
{  
    Serial.begin(9600);  
}
```

```
void loop()  
{  
    //segnale misurato (prodotto sintenticamente ora nel programma)  
    s = a1*sin(omega1*time) + a2*sin(omega2*time);  
  
    //uso del filtro del primo ordine  
    f = theta*f + (1-theta)*s;  
  
    if( (s <= -0.1) && (s >= 0.1))  
    {  
        digitalWrite(pinLed, HIGH);  
    }  
    else  
    {  
        digitalWrite(pinLed, LOW);  
    }  
}
```

```
}
```

```
Serial.println(s);
```

```
delay(1);  
time = time + 0.001;
```

```
}
```

```
////////////////////////////////////
```

Esercizio 17 Dic 2019

1) PWM

```
#define pinMotor 9
```

```
void setup()
```

```
{  
}
```

```
void loop()
```

```
{  
  analogWrite(pinMotor,155);  
  delay(1000); // Wait for 1000 millisecond(s)  
  digitalWrite(pinMotor, 0);  
  delay(1000); // Wait for 1000 millisecond(s)  
}
```

2) Azionamento di un motore elettrico in corrente continua.

```
/* LA VELOCITA' DI ROTAZIONE DEL MOTORE (OMEGA) E' PARI A
```

```
OMEGA = 3*TENSIONEaIMOTORE
```

La tensione ai capi del motore è la tensione letta dal multimetro
Cerca il valore del duty cycle affinché la velocità
del motore è pari a 3 radianti al secondo ($\omega = 3$).

*/

// TensioneAIMotore = $\Omega/3 = 3/3 = 1V$
//devo cercare il valore del duty cycle tale che la tensione
//letta ai capi del motore è circa 1V

#define pinMotor 9
float tensioneAIMotore = 0;

// FILTRO
float theta = 0.99;
float Vfiltrata = 0.0;

void setup()
{
 pinMode(13,OUTPUT); //preparo il piedino associato al led
 Serial.begin(9600); //abilito la porta seriale
}

void loop()
{
 //lettura del sensore
 tensioneAIMotore = 5*analogRead(A0)/1023.0;

Vfiltrata = theta*Vfiltrata + (1-theta)*tensioneAIMotore;

analogWrite(pinMotor,127); //definire il duty cycle del piedino PWM numero 9
Serial.println(Vfiltrata); //stampo il valore letto

/*

digitalWrite(13,HIGH);
delay(2000); // Wait for 1000 millisecond(s)

digitalWrite(pinMotor, 0); //apro il circuito
digitalWrite(13,LOW);
delay(2000); // Wait for 1000 millisecond(s)

*/

```
delay(1);  
}
```

