# 03_Python_3_Pacchetto_livello_3_7_modello_ISO

### parte 1

Nella prima fase dell'attività abbiamo fatto delle considerazioni preliminari riguardanti la struttura del pacchetto e della parte dati, contenente network id, mittente, destinatario e tipo.

La costruzione del pacchetto in C costruiti tramite struct, la ricezione e la decodifica del pacchetto in C, la costruzione ed invio del pacchetto in Python, la ricezione e decodifica del pacchetto in Python.

### parte 2

Nella seconda fase abbiamo visto per punto per punto la struttura dei 4 programmi che avremmo dovuto realizzare nella fase successiva.

### parte 3

**Fase 3.1:** In questa prima fase della parte 3 abbiamo scritto un programma per inviare i dati ricevuti dal sensore secondo la struttura del pacchetto trattata nella parte 1.

sketch sensore arduino:

```
#define ID "BE"
#define MITTENTE "M001"
#define DESTINATARIO "D031"
#define TIPO "S1"
#define VUOTO "----------------"

struct pacchettoS1 {
    char id[2];
    char mittente[4];
    char destinatario[4];
    char tipo[2];
    char valoreSensore[4];
    char vuoto[16];
  };

void setup()
{
  Serial.begin(9600);
}
void loop()
{
  int num = analogRead(A0);
  char s[5];
  sprintf(s, "%04d", num);

  struct pacchettoS1 msg;
  memcpy(msg.id, ID, sizeof(msg.id));
  memcpy(msg.mittente, MITTENTE, sizeof(msg.mittente));
  memcpy(msg.destinatario, DESTINATARIO, sizeof(msg.destinatario));
  memcpy(msg.tipo, TIPO, sizeof(msg.tipo));
  memcpy(msg.valoreSensore, s, sizeof(msg.valoreSensore));
  memcpy(msg.vuoto, VUOTO, sizeof(msg.vuoto));

  Serial.write((byte *)&msg, sizeof(msg));

  delay(1000);
}
```

**Fase 3.2:** In questa seconda fase abbiamo scritto il primo programma python con il compito di ricevere i dati dal sensore e decodificare il pacchetto ricevuto, scartandolo nel caso in cui il pacchetto non sia destinato a noi o abbia un id diverso dal nostro. (video funzionamento in allegato).

programma python ricezione dati sensore:

```
import serial
import struct
IDCORRETTO = "BE"
DESTINATARIOCORRETTO = "D031"

arduino = serial.Serial('COM3', 9600)

print('inizio ricezione dei dati')
while True:
    val = arduino.read(32)
    pack = struct.unpack("2s 4s 4s 2s 4s 16s", val)
    id=pack[0].decode()
    mittente=pack[1].decode()
    destinatario=pack[2].decode()
    tipo=pack[3].decode()
    valoreSensore=pack[4].decode()
    vuoto=pack[5].decode()
    if (id==IDCORRETTO)and(destinatario==DESTINATARIOCORRETTO):
        print("id e destinatario corretti")
        print(valoreSensore)
    else:
        print("pacchetto scartato")
```

**Fase 3.3:** In questa terza fase dell'attività abbiamo modificato il programma python in modo da chiedere i dati, creare e inviare il pacchetto contenente i dati ad arduino.

programma python invio comandi al motore:

```
import serial;
import struct;
ID=b"BE"
MITTENTE=b"M001"
DESTINATARIO=b"D031"
TIPO=b"A1"
VUOTO=b"----------------"

arduino = serial.Serial('COM3', 9600)

print('inizio invio dei dati')
while True:
    d = input("direzione A/I/S: ")
    if (d=="S"):
        v=0
    else:
        v = input("velocità: ")
        VELOCITA=str(v).zfill(3).encode()
    DIREZIONE = str(d).zfill(1).encode()
    pack=struct.pack("2s 4s 4s 2s 1s 3s 16s",ID,MITTENTE,DESTINATARIO, TIPO, DIREZIONE, VELOCITA, VUOTO)
    arduino.write(pack)
    print(pack)
```

**Fase 3.4:** In ques'ultima fase abbiamo riscritto lo sketch di arduino per fare in modo che questo riceva il pacchetto, lo decodifichi, controlli che l'id e il destinatario siano corretti e infine invii i comandi al motore. (video funzionamento in allegato).

sketch arduino motore:

```
#define ID "BE"
#define DESTINATARIO "D031"

struct pacchettoA1
{
  char id[2];
  char mittente[4];
  char destinatario[4];
  char tipo[2];
  char direzione[1];
  char velocita[3];
  char vuoto[16];
};
```

```
void setup()
{
  Serial.begin(9600);
  pinMode(3, OUTPUT);
  pinMode(5, OUTPUT);
}

void loop()
{
  struct pacchettoA1 msg;
  if (Serial.available())
  {
    Serial.readBytes((byte*) &msg, sizeof(msg));
    int controlloId = memcmp(ID, msg.id, 2);
    int controlloDest = memcmp(DESTINATARIO, msg.destinatario, 4);
    int n = sizeof(msg.velocita);
    char vel[4];
    if (controlloId == 0 && controlloDest == 0)
    {
      memcpy(vel,msg.velocita,sizeof(msg.velocita));
      vel[3] = '\0';
      int velocita = atoi(vel);
      String direzione = (String)msg.direzione;
      if (memcmp("A",msg.direzione, 1) == 0)
      {
        digitalWrite(3, LOW);
        digitalWrite(5, HIGH);
        digitalWrite(9, velocita);
      }

      if (memcmp("I",msg.direzione, 1) == 0)
      {
        digitalWrite(3, HIGH);
        digitalWrite(5, LOW);
        digitalWrite(9, velocita);
      }
      if (memcmp("S",msg.direzione, 1) == 0)
      {
        digitalWrite(3, LOW);
        digitalWrite(5, LOW);
        digitalWrite(9, 0);
      }
    }
  }
}
```