**Pacchetto su RF24: conversione dei programmi da seriale a RF24**

*(Parte Arduino)*

1. **Considerazioni preliminari**

*Lo scopo di questa attività è modificare i programmi realizzati nella fase* ***“Pacchetto Livello 3 – seriale”****, in modo che facciano uso del modulo radio* ***nRF24L01+*** *che è già stato analizzato ed utilizzato nell’attività* [*Arduino e RF24*](https://www.vincenzov.net/tutorial/elettronica-di-base/Trasmissioni/laboratorio/RF24a.htm) *in Telecomunicazioni. Fate riferimento a tale esercitazione per il cablaggio e il settaggio dei parametri.*

*L’attività è svolta in coppia, con le stesse coppie delle attività di installazione Raspberry. La programmazione Arduino avverrà comunque, per maggior comodità, dall’IDE sotto Windows.*

*La struttura dei programmi è esattamente la stessa sia per Arduino che per Python, solo cambiano, oltre ovviamente al cablaggio, le operazioni di settaggio iniziale e i metodi di lettura/scrittura.*

*Risulterà invece un po’ più complessa la fase di sviluppo e testing: per questa ragione si procederà in modo incrementale, testando un componente alla volta, in quest’ordine:*

1. *Sviluppo del programma Arduino di invio per il sensore: il pacchetto verrà inviato dal vostro programma ma ricevuto da un programma centralizzato messo a disposizione dall’insegnante (pipe="00001")*
2. *Sviluppo del programma Arduino di ricezione per l’attuatore: il pacchetto verrà inviato da un programma centralizzato messo a disposizione dall’insegnante e ricevuto dal vostro programma (pipe="00002", id=”EP”, mittente=”S730”, destinatario=”P438”, velocità variabile fra 100 e 250, direzione variabile ‘A’ o ‘I’)*
3. *Sviluppo del programma Python di ricezione per il sensore: il pacchetto verrà inviato dal vostro programma su Arduino (già testato nella fase 1) e ricevuto dal vostro programma Python su Raspberry*
4. *Sviluppo del programma Python di invio per l’attuatore: il pacchetto verrà inviato dal vostro programma Python su Raspberry e ricevuto dal vostro programma su Arduino (già testato nella fase 2)*

*Per le fasi 1 e 2 verranno date alcune informazioni sul settaggio dei parametri mentre per tutte le altre informazioni fate riferimento all’esercitazione di Telecomunicazioni*

*Per le fasi 3 e 4 verranno fornite le informazioni di cablaggio su Raspberry e l’uso delle funzioni di libreria. Anche in questo caso la struttura dei programmi rimane inalterata, sostituendo soltanto i metodi da chiamare. Queste fasi sono illustrate nell’attività seguente* ***“Pacchetto Livello 3 - RF24 – Raspberry”***

1. **Arduino – sensore**

*Definizione costanti e pacchetto*

#define ID "AB" // nome attribuito alla rete logica

#define TIPO "S1" // tipo di pacchetto

#define MITTENTE "S730" // nome sensore (a scelta)

#define DESTINATARIO "P001" // nome Python ricevente (a scelta)

#define WRITINGPIPE "00001" // pipe di invio

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

struttura del pacchetto di 32 byte - tipo sensore S1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct pacchettoS1 {

char id[2];

char mittente[4];

char destinatario[4];

char tipo[2];

char valoreSensore[4];

char vuoto[16];

};

struct pacchettoS1 msg;

*Settaggi (la Serial è utilizzata a scopo di debugging)*

#include <RF24.h>

RF24 radio(7, 8); // CE, CSN

void setup() {

Serial.begin(9600);

radio.begin();

radio.setPALevel(RF24\_PA\_MIN);

radio.setPayloadSize(32);

radio.setDataRate(RF24\_2MBPS);

radio.openWritingPipe((byte \*)WRITINGPIPE);

radio.stopListening();

}

*Invio (la Serial è utilizzata a scopo di debugging)*

// scrive su seriale a scopo di debugging

Serial.write((char \*)&msg, sizeof(msg));

Serial.println();

// invia su RF24

radio.write((char \*)&msg, sizeof(msg));

1. **Arduino – attuatore**

*Definizione costanti e pacchetto*

#define MIO\_ID "AB" // nome attribuito alla rete logica

#define MIO\_TIPO "A1" // tipo di pacchetto

#define MIO\_INDIRIZZO "A328" // nome attribuito all'attuatore

#define READINGPIPE "00002" // pipe di ricezione

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

struttura del pacchetto di 32 byte – tipo A1

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

struct pacchettoA1 {

char id[2];

char mittente[4];

char destinatario[4];

char tipo[2];

char direzione[1];

char velocita[3];

char vuoto[16];

};

struct pacchettoA1 msg;

*Settaggi (la Serial è utilizzata a scopo di debugging)*

#include <RF24.h>

RF24 radio(7, 8); // CE, CSN

void setup() {

Serial.begin(9600);

radio.begin();

radio.setPALevel(RF24\_PA\_MIN);

radio.setPayloadSize(32);

radio.setDataRate(RF24\_2MBPS);

radio.openReadingPipe(0, (byte \*) READINGPIPE);

radio.startListening();

}

*Ricezione (la Serial è utilizzata a scopo di debugging)*

if (radio.available() ) { // controlla che ci siano dati

radio.read((char \*) &msg, sizeof(msg)); // legge il pacchetto

Serial.println((char \*)&msg); // scrive su seriale a scopo di // debugging