

Sistemi distribuiti e intelligenza artificiale distribuita

Modellazione e prototipazione di una centralina per il rilevamento di gas naturale, gestito da agenti autonomi Java, interfacciati su uno spazio di tuple

Di Blasi Fabrizio

Introduzione

- Il progetto presentato riproduce in piccola scala una possibile porzione di una condotta per il trasporto di gas GPL/propano
 - Il modellino è provvisto di tre paratie di chiusura, due simulate da due coppie di LED verdi e rossi ed una comandata da un servomotore, e due sensori di GAS posti sulla sommità della condotta.
 - I sensori e gli attuatori sono comandati da due schede Arduino Uno R3.
-

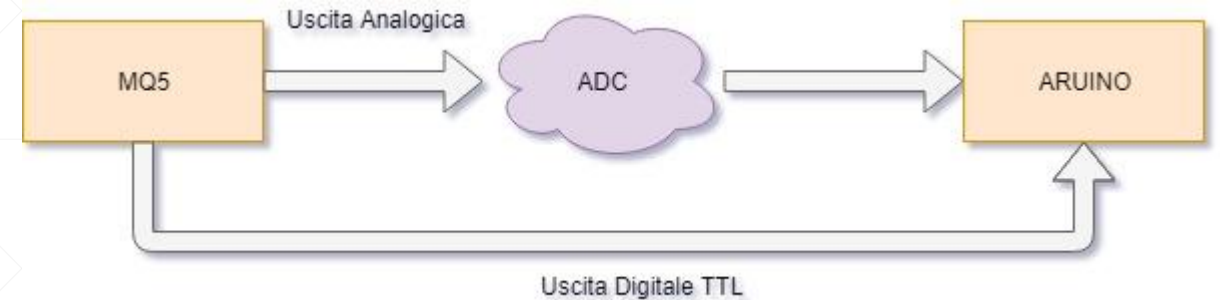
Componenti adottati

- Arduino UNO R3
- Sensore di gas MQ5
- Servomotore
- LED (rosso e verde)

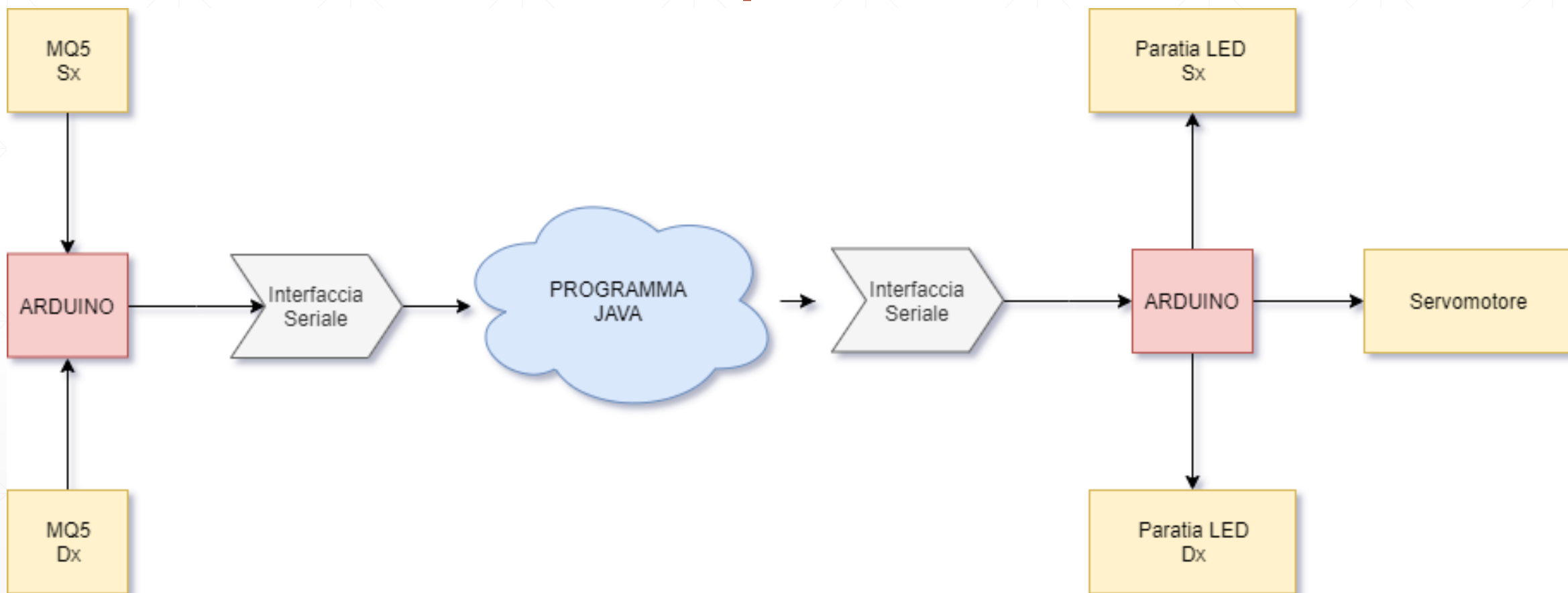


Sensore Mq5

- Il sensore comunica con due segnali:
- Digitale TTL
 - Segnale basso se rileva GAS
 - Segnale alto se non c'è rilevamento
- Uscita Analogica
 - Misurazione del livello di gas
 - È possibile ottenerla anche quando l'uscita TTL non è nulla
- In base al segnale TTL si invia una specifica keyword al BRIDGE



Schema a Blocchi complessivo

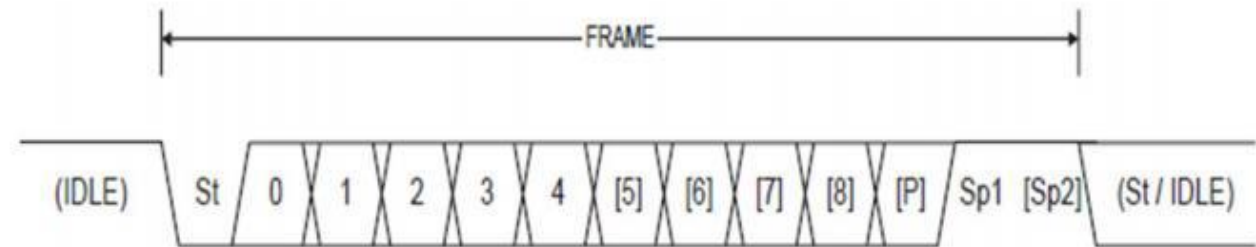


Descrizione dello schema a blocchi

- Il programma JAVA si interfaccia con i due microcontrollori attraverso il protocollo seriale
 - Il microcontrollore ha il compito di misurare la presenza, o meno, di gas, esso effettua una comunicazione monodirezionale con il programma di controllo JAVA, il quale, in base all'andamento dei valori delle misurazioni, sceglierà quale comando fornire al secondo microcontrollore che azionerà gli attuatori.
-

Protocollo Seriale

- Baud rate fissata a 9600
- Non c'è bisogno dell'invio del clock
- Pacchetti di 9 bit
 - 8 di dato
 - 1 di controllo (opzionale)



St Start bit, always low.

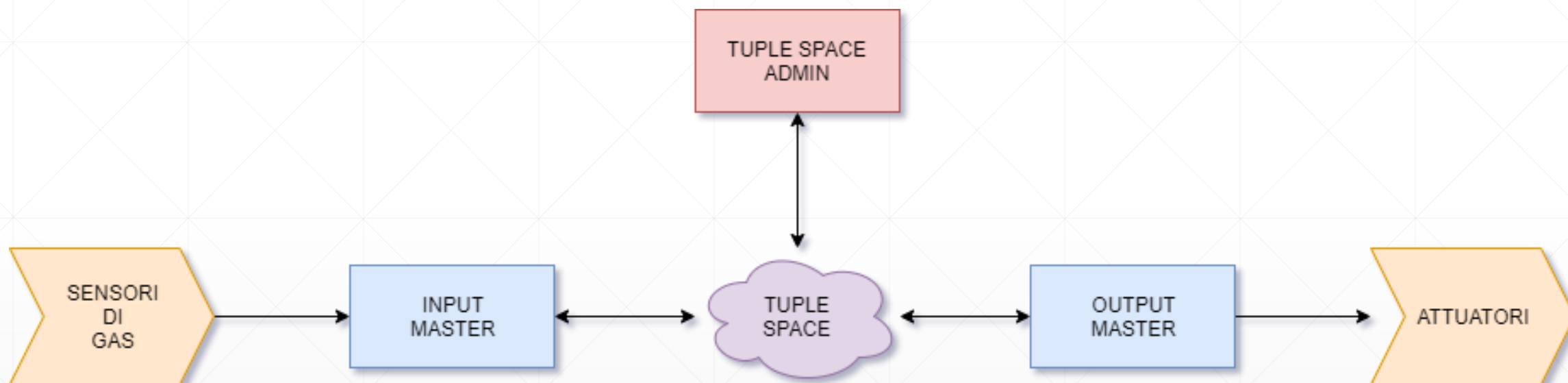
(n) Data bits (0 to 8).

P Parity bit. Can be odd or even.

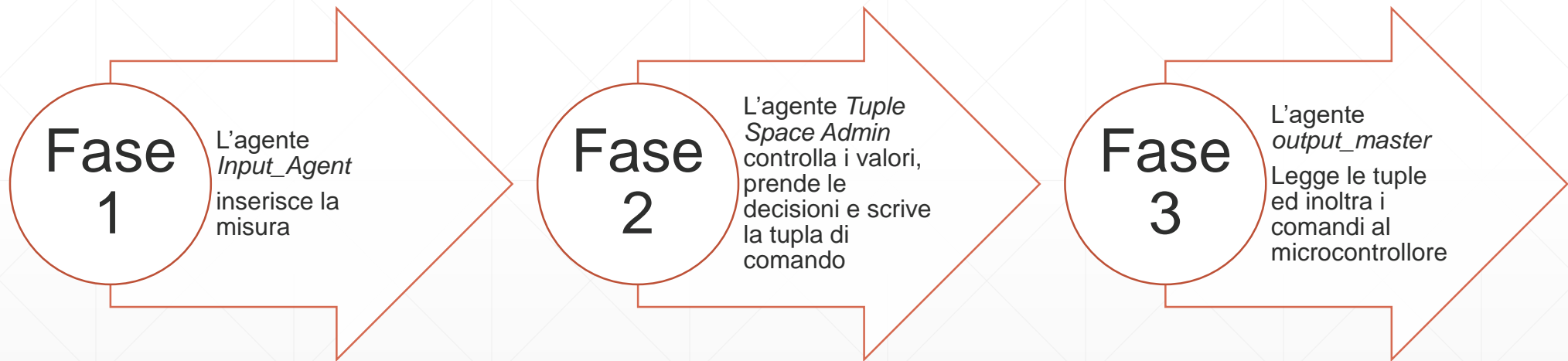
Sp Stop bit, always high.

IDLE No transfers on the communication line (RxD or TxD). An IDLE line must be high.

Struttura del programma Java



Funzionamento del software Java



Tipi di tuple scambiate nello spazio

- *Input Agent :*

- Preleva i valori da linea seriale e scrive le tuple che contengono le misurazioni di GAS, che sono così strutturate

< id sensore ['S' / 'D'] (char), valore misura (int) >

- *Tuple Space Agent:*

- Nel momento in cui rileva un valore di gas non idoneo, inserisce una tupla così fatta:

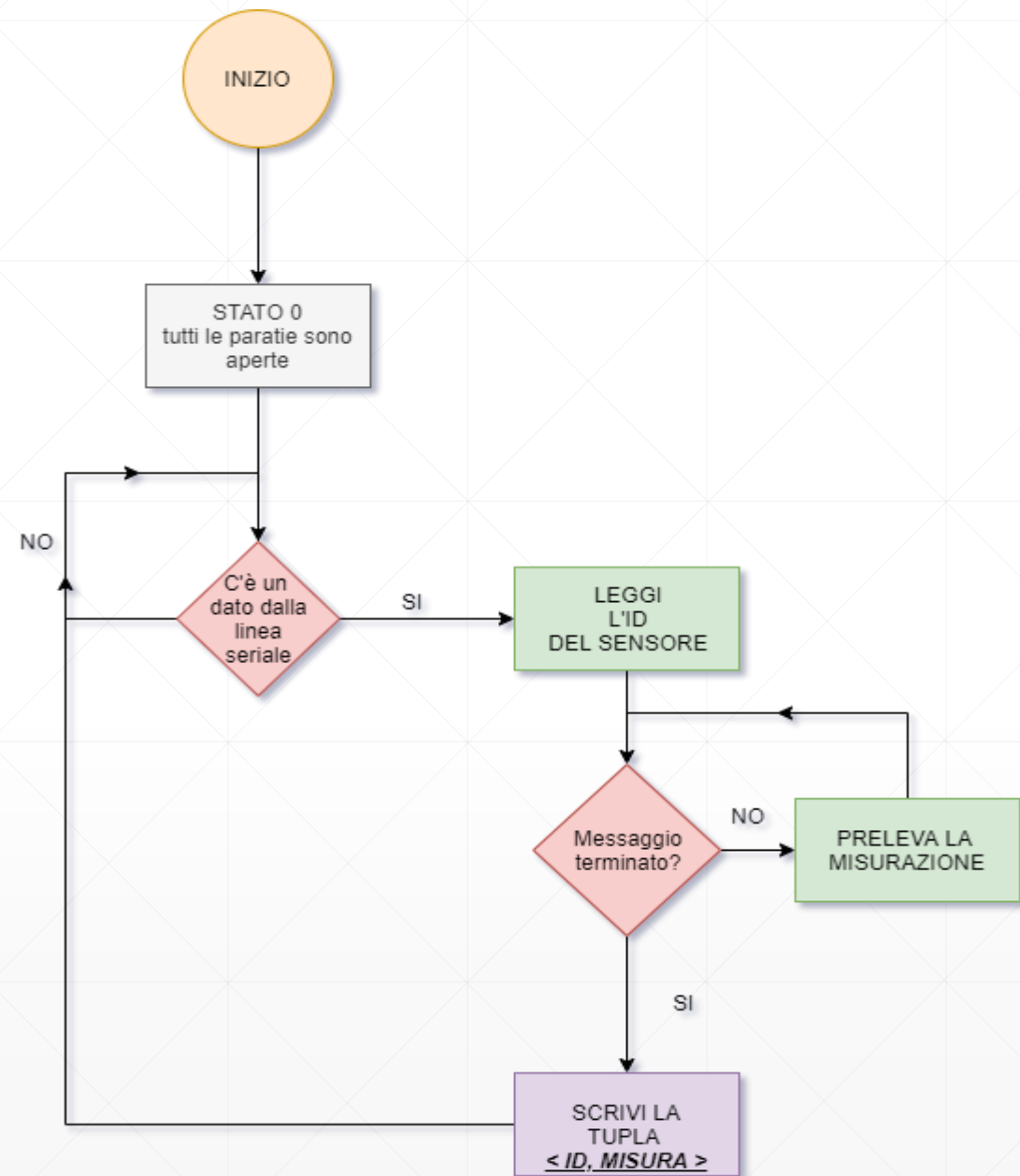
< id sensore ['S' / 'D'] (char), comando ['C' / 'O'] (char) >

- *Output Agent:*

- Si mette in ascolto di tuple aventi entrambi i campi di tipo *char* (quelle inserite da *Tuple space agent*) ed inoltrerà tali comandi su linea seriale al microcontrollore che comanda gli attuatori
-

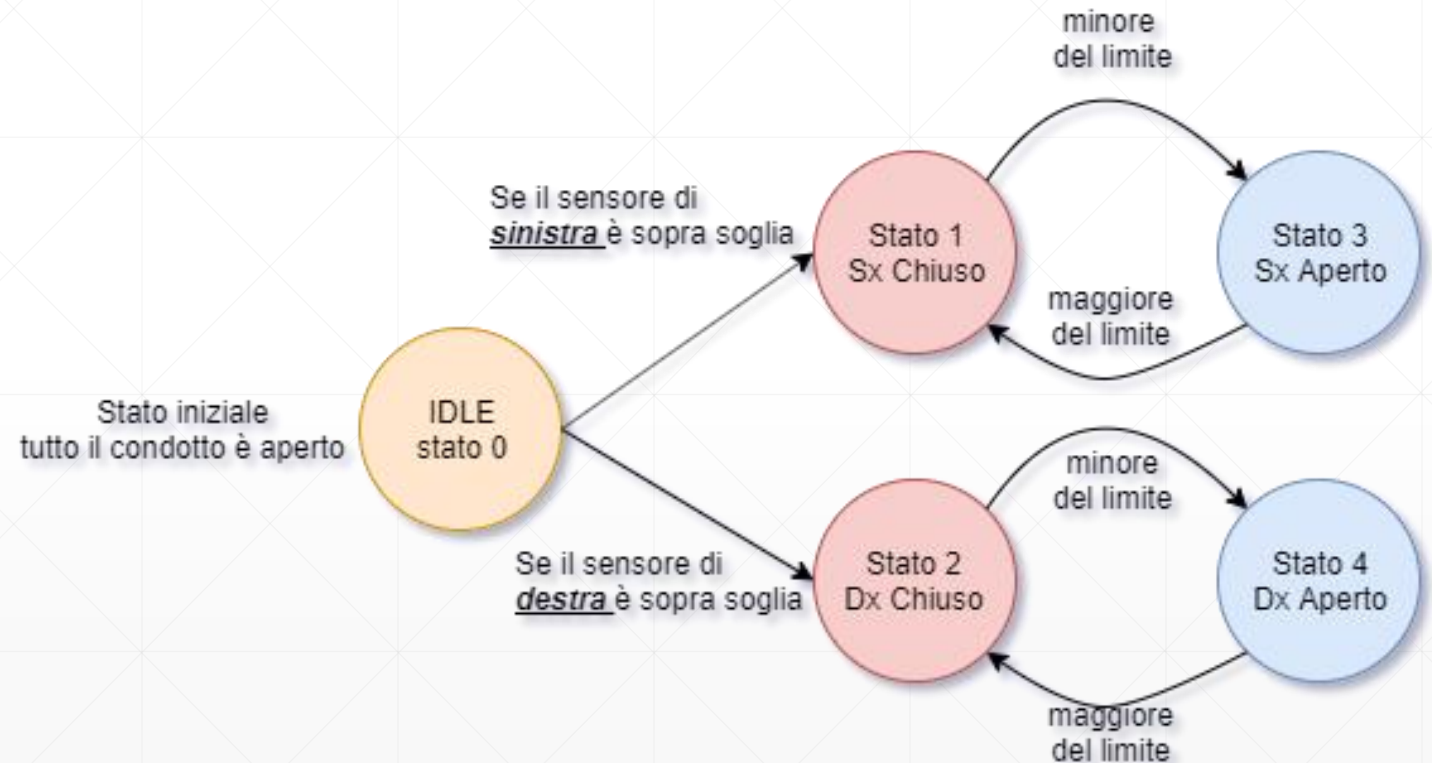
Input Agent

- Tale agente preleva le misurazioni dal microcontrollore
- Nel momento che arriva un byte, controlla se il dato coincide con l'id e lo memorizza
- Finché non legge il terminatore di messaggio codificato con il carattere '_' continuerà a prelevare la misura
- Finita la ricostruzione del messaggio scrive la tupla :
< id sensore ['S' / 'D'] (char), valore misura (int) >
- Poi si mette in attesa di un nuovo byte in arrivo dalla linea seriale



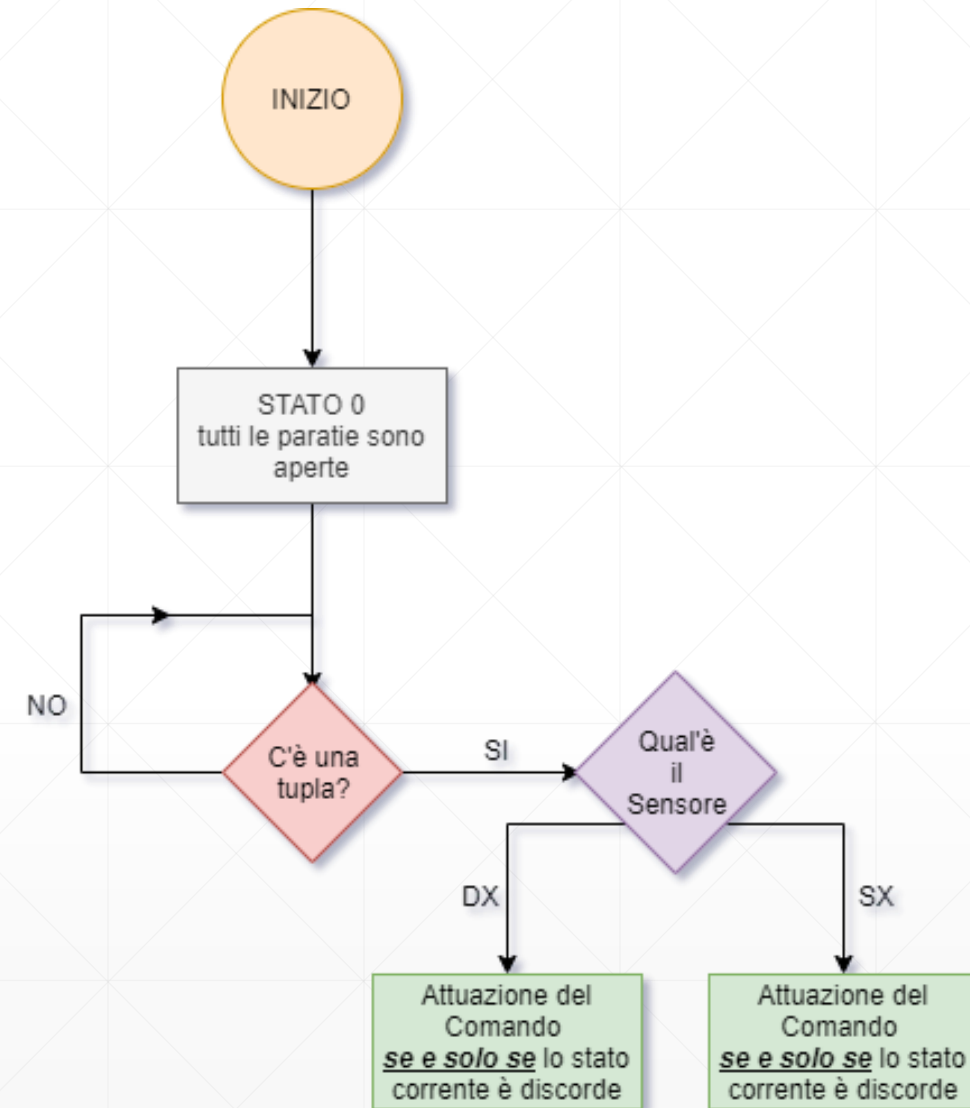
Tuple Space Agent

- È l'agente più importante, ha il compito di amministrare lo spazio di tuple.
- Ha una logica interna a stati
- Nell'istante iniziale tutti i sistemi sono funzionanti correttamente
- Quando legge un valore sopra alla soglia cambia lo stato
- Lo stato è diverso in base a quale sensore ha dato l'allarme
- in ogni stato il sistema andrà a scrivere la tupla <id, misura>



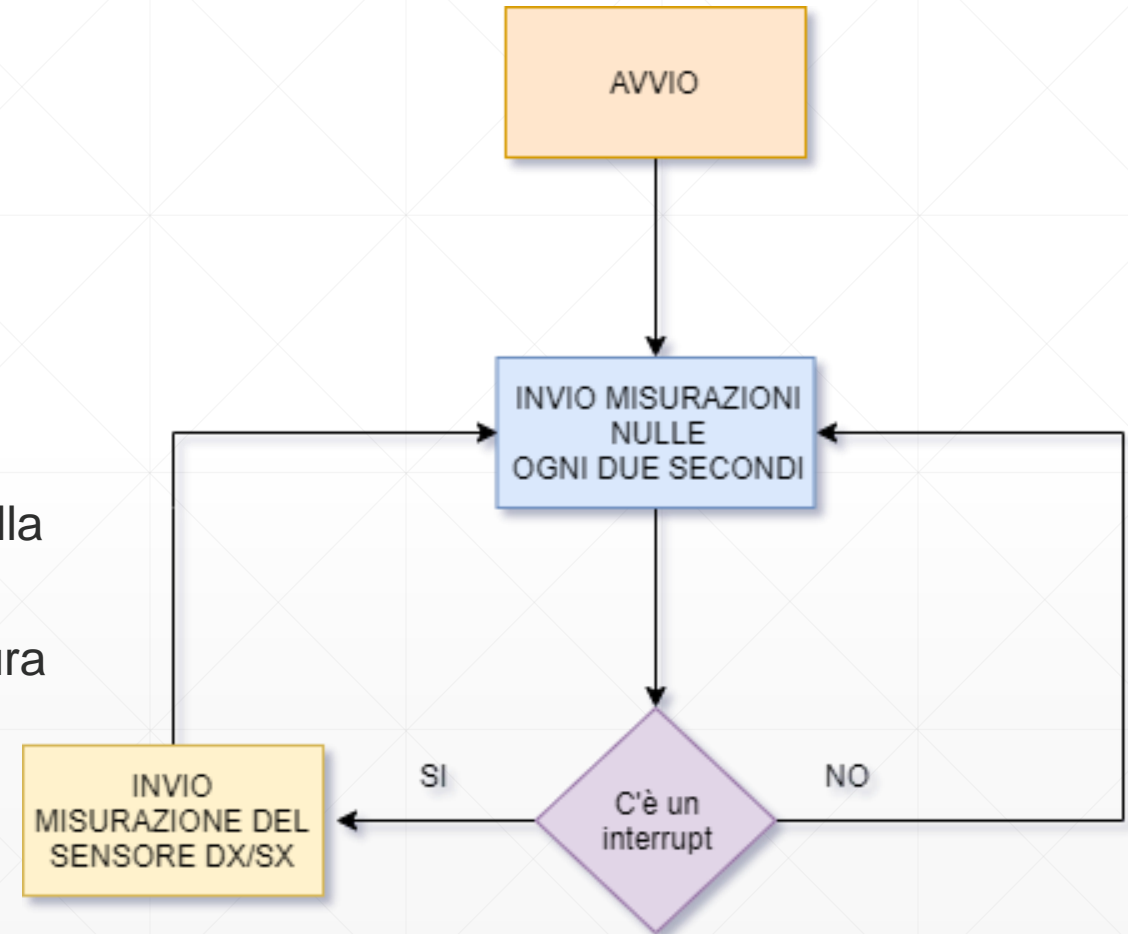
Output Agent

- Tale agente tiene in memoria lo stato della condotta
- Si mette in ascolto sullo spazio delle tuple e sottoscrive il suo interesse a tutte quelle così formate :
< id sensore ['S' / 'D'] (char), comando ['C' / 'O'] (char) >
- Andrà ad inoltrare il messaggio al microcontrollore solamente se leggerà un comando in contraddizione con lo stato attualmente in memoria



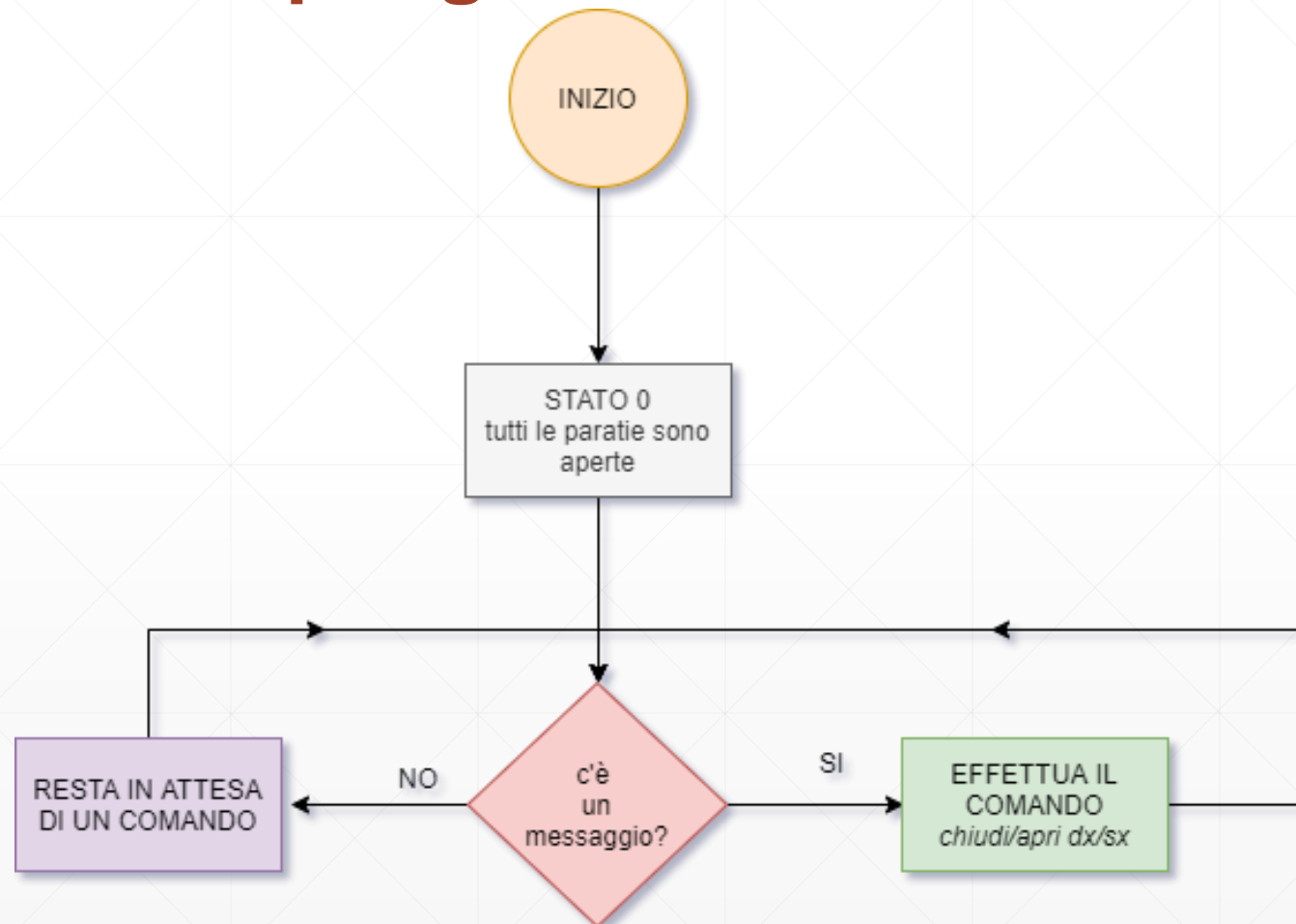
Funzionamento del software per il rilevamento del gas

- Due caratteristiche principali:
 - Invio del rilevamento del Gas con una particolare keyword anteposta al valore della misura che può essere:
 - S
 - D
 - In condizioni normali invia una misura nulla
 - Se capta del GAS arriva un interrupt che invia istantaneamente il valore della misura



Funzionamento del software per gli attuatori

- Inizialmente tutti i sistemi sono aperti e funzionanti, privi di interruzioni
- Il microcontrollore resta in attesa di un comando proveniente dall'agente *output agent*
- Se arriva un comando va lo analizza e lo esegue. Nel mentre, il microcontrollore tiene traccia dello stato delle paratie in una variabile
- La paratia centrale si va ad aprire solamente se entrambi i lati hanno il led *verde* acceso



Conclusioni

- Grazie all'uso dello spazio di tuple è possibile disaccoppiare gli agenti dal loro compito, inoltre non è necessario che gli agenti si conoscano a vicenda.
 - Gli agenti devono solamente conoscere ed avere il riferimento all'oggetto dello spazio di tuple e sapere la logica con il quale vengono scritte le tuple.
 - In questo modo gli agenti possono essere dislocati nella rete.
 - La nomenclatura degli *ID* utilizzata nel progetto non è propriamente flessibile ma basterebbe modificarlo con un *ID* numerico in modo da rendere più dinamico il tutto
 - Utilizzando un ID numerico, però, si avrebbe la necessità di avere un microcontrollore per ogni dispositivo, che sia attuatore o sensore. In questo modo si migliora la gestione
 - Gli agenti lavorano sulla stessa macchina perciò non è stato implementato un servizio di naming per l'oggetto che gestisce lo spazio di tuple
 - Andando verso un ottica più distribuita di questa presentata, tale servizio assume un'importanza vitale
-

Grazie per l'attenzione
