

Diario di lavoro

Luogo	Canobbio
Data	14.02.2020

Lavori svolti

Oggi ho scritto la classe **SerialResponse** la quale estende **SerialCommunication** e contiene i tipi di risposta possibili, **OK** e **ERROR**.

Poi ho creato un interfaccia **SerialInputCommand** che viene utilizzata per i comandi che invia L'arduino al raspberry, la quale contiene il metodo **response()** che serve per creare la risposta al comando. Questa interfaccia verrà implementata nelle sotto classi di **SerialCommand**, relativi hai comandi che l'Arduino invia al Raspberry.

Dopo di che ho implementato **SerialEchoCommand**.

Poi ho iniziato a sistemare il codice relativo all'Arduino per l'utilizzo della scelta fra comando e risposta. Per questo motivo ho provato la connessione, la quale si blocca perché il metodo **getInstruction()** ritorna sempre 0.

Dopo aver risolto il problema ho eseguito un test del comando echo, manualmente (senza unit testing). Nella classe **App**.

```
SerialEchoCommand sc = new SerialEchoCommand("HelloWorld!".getBytes());
sc.write(output);
```

```
SerialCommunication serialCommunication = new SerialCommunication();
serialCommunication.readUntilEnd(input);
System.out.println((char) serialCommunication.getSequence());
System.out.println(new String(serialCommunication.getMessage()));
```

Il codice è stato inserito nella classe già presente. Il quale dovrebbe stampare sulla linea di comando il carattere del comando (**o**) ed il testo inviato (**HelloWorld!**).

Problemi riscontrati e soluzioni adottate

Quando richiama il metodo arduino **getInstruction()** ritornava sempre 0, quindi analizzando il codice mi sono accorto che trasformando il codice dall'utilizzo dei **char** all'utilizzo dei **byte** nel metodo seguente:

```
/**
 * Read instruction from serial.
 */
boolean serialRead(byte sequenceType) {
    Serial.readStringUntil(sequenceType);

    // read from serial line
    String s = Serial.readStringUntil(SERIAL_END_OF_COMMAND);

    // if correctly read exists
    if (s.length() > 0) {
        // set command byte to instruction variable
        instruction = (byte) s[0];

        // set command value to instruction value variable
        instructionValue = s.substring(1);
```

```
    return true;  
}  
  
    return false;  
}
```

Alla linea di assegnazione della variabile mancava il casting, **(byte)**.

Punto della situazione rispetto alla pianificazione

Sono in linea con la pianificazione. Attività 12

Programma di massima per la prossima giornata di lavoro

Creare la thread di controllo della porta seriale ed integrarla con il webserver.