Diario di lavoro

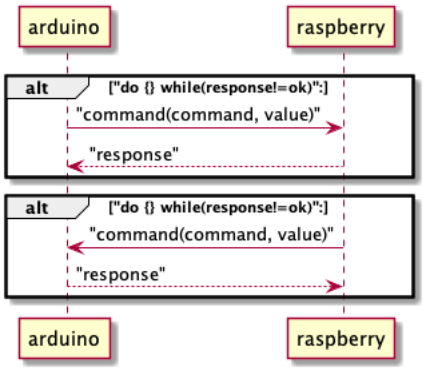
Luogo	Canobbio
Data	07.02.2020

Lavori svolti

Oggi ho deciso di provare a fare una piccola implmenetazione della comunicazione seriale, con java, quindi per prima cosa progetterò la comunicazione seriale. Poi la proverò ad implementare ed infine provero a collegarla ad un web server.

Il protocollo di comunicazione:

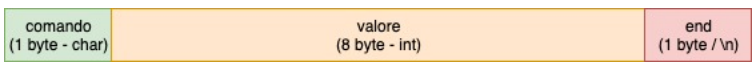
Siccome il codice dell'Arduino è basato principalmente su un metodo che viene eseguito in ciclo, finché esso è alimentato, per ogniuno di questi cicli l'Arduino invierà e riceverà un comando.



Come si puo vedere nel diagramma soprastante, il comando sia di recezione che di invio viene eseguito finche non riceve un messaggio di risposta "**OK**".
I possibili comandi sono elencati sotto:

Comando	Nome	Direzione	Tipo di dato	Dimensione	Note
n	null	←→	-	-	messaggio vuoto – no action
f	frequence	←	int	0-25000	cambio frequenza
s	status	←	boolean	-	on/off generator
r	remote	→	-	-	remote changed status
m	mic	→	-	-	valore microfono
o	ok	←→	-	-	messaggio ricevuto correttamente
e	error	←→	-	-	errore di comunicazione
d	decibel	←	Int	0-200	cambio di decibel per microfono

I messaggi, hanno la seguente struttura.



Dopo aver progettato il protocollo ho fatto un indagine su quale libreria sia la migliore per effettuare la connessione tramite porta seriale con Java. Ne ho trovate 2:

- jSerialComm di (**https://fazecast.github.io/jSerialComm/**)
- RXTX (**http://rxtx.qbang.org/wiki/index.php/Download**)

Per utilizzare la prima libreria basta gradle, mentre per la seconda bisogna scaricare ed inserire una libreria nella path delle librerie di Java. Per facilitarmi il lavoro ho deciso di utilizzare la prima.

Per iniziare ho scritto un programma per arduino che scrive la stringa "**HelloWorld!**" sulla seriale, per testare l'invio dei dati.

```
// setup arduino
void setup() {
  // open channel at 9600 bit/s
  Serial.begin(9600);
  // send communication start char
  Serial.write(42);
}

// loop of the arduino
void loop() {
  Serial.write("HelloWorld!\n");
}
```

Dopo di che ho scritto il codice che legge dal lato Java. Importando la libreria con gradle:

```
dependencies {
    // This dependency is found on compile classpath of this component and consumers.
    implementation 'com.google.guava:guava:27.0.1-jre'

    // Use JUnit test framework
    testImplementation 'junit:junit:4.12'
       // https://mvnrepository.com/artifact/com.fazecast/jSerialComm
       compile group: 'com.fazecast', name: 'jSerialComm', version: '2.5.3'
}
```

Codice:

```
public class App {
    public static final byte INIT = 42;

    public static void main(String[] args) {
        SerialPort serialPort =  SerialPort.getCommPort("/dev/tty.usbmodem14101");
        serialPort.setComPortTimeouts(SerialPort.TIMEOUT_READ_BLOCKING, 100000000,
10000000);
        System.out.println(serialPort.openPort());
        InputStream input = serialPort.getInputStream();

        int read;
        try {
            while ((read = input.read()) != 0) {
                byte c = (byte)(0x000000FF & read);

                if (c == INIT) {
                    System.out.println("Initialized");
                }
                System.out.print((char)c);

            }
        } catch (IOException ioe) {
            ioe.printStackTrace();
            System.out.println(ioe.getMessage());
        }
    }
}
```

Inizialmente ho avuto un problema, dopo pochi byte di trasmissione essa veniva interrotta con il

seguente errore:

```
com.fazecast.jSerialComm.SerialPortTimeoutException: The read operation timed out
before any data was returned.
      at
com.fazecast.jSerialComm.SerialPort$SerialPortInputStream.read(SerialPort.java:1427)
      at serial.acc.java.App.main(App.java:22)
The read operation timed out before any data was returned.
```

Cercando in internet quale potrebbe essere l'errore ho trovato sul seguente link **https://forum.arduino.cc/index.php?topic=613381.0** il problema è che quando viene aperto un monitor seriale con l'arduino quest'ultimo si resetta. Quindi bisogna lascargli il tempo. Siccome la libreria non riceve nessun dato dal metodo read() dopo diverso tempo interrompe il processo.
Quindi cercando nella documentazione della libreria ho trovato sulla pagina **https://github.com/Fazecast/jSerialComm/wiki/Modes-of-Operation** una possibile soluzione. Utilizzare il metodo **setComPortTimeouts()**, per il quale ho cercato la documentazione, nella documentazione il suo funzionamento. Per poi aggiungere al codice la seguente linea di codice, che mi permette di aumentare il time-out del metodo **read()**.

```
serialPort.setComPortTimeouts(SerialPort.TIMEOUT_READ_BLOCKING, 100000000, 10000000);
```

Problemi riscontrati e soluzioni adottate

Un problema relativo alla comunicazione seriale con l'arduino, descritto e risolto sopra.

Punto della situazione rispetto alla pianificazione

Sono ancora indietro di 12 rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Finire di testare il collegamento seriale ed integrarlo con il mini-server web.