Test 1: School

1. Setup:

1) Upload your code to the first School Board Atmega (Atmega1)
2) Upload the code “Complete_school_boardTest1” to the second Atmega (Atmega2)
3) In the code “sketch_oct06aMasterTest1”, Line 14 change the value of “I2CAdressSchoolToTest” to match the i2c address of your school.

```
const int I2CAdressSchoolToTest = 1; //I2C adress of the board we want
to test.
```

4) Upload the code “sketch_oct06aMasterTest1” to the Arduino Uno (remember to change the board type to ‘Arduino Uno’ in the IDE
5) Insert the 2 School board Atmegas in the testbeds of the soldered testing board.
6) Connect the Arduino Uno to the Testing board (see connection in annexe)
7) Open the Serial monitor.

Note : You may have to reset the testing board before testing by disconnecting and reconnecting the 3V3.

2. Expected result:

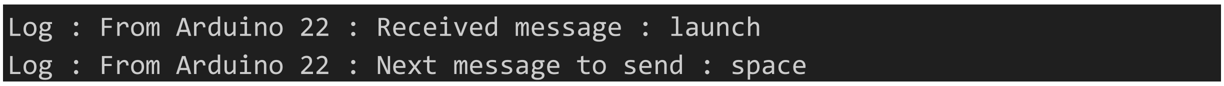
<u>During Round 1:</u>

- **<u>Test 1.1</u>**: The Atmega 1 must calibrate its photoresistor at the beginning of the round and send the log to the master:

```
############################################
beginning of the round 1 :
degraded acceptance test
############################################
Request Arduino 1 Init LDR
Request Arduino 22 Init LDR
Log : From Arduino 1 : Calibrating the LDR ...
Log : From Arduino 22 : Calibrating the LDR ...
Log : From Arduino 1 : LDR Calibration Done.
Log : From Arduino 1 : Max Luminosity = 1
Log : From Arduino 1 : Mean Luminosity = 0.25
Log : From Arduino 1 : Threshold Value = 50
Log : From Arduino 22 : LDR Calibration Done.
Log : From Arduino 22 : Max Luminosity = 7
Log : From Arduino 22 : Mean Luminosity = 6.35
Log : From Arduino 22 : Threshold Value = 56
```

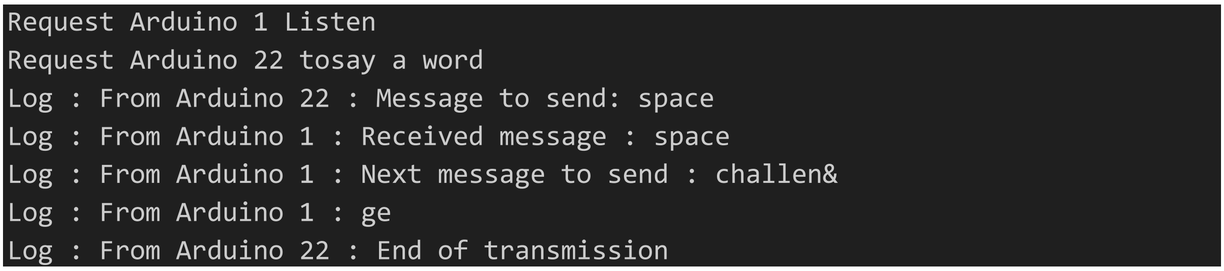
- **<u>Test 1.2:</u>** The Atmega1 must send the first word and Atmega2 must receive it correctly:

```
Request Arduino 22 Listen
Request Arduino 1 tosay a word
Log : From Arduino 1 : Message to send: launch
Log : From Arduino 1 : End of transmission
```



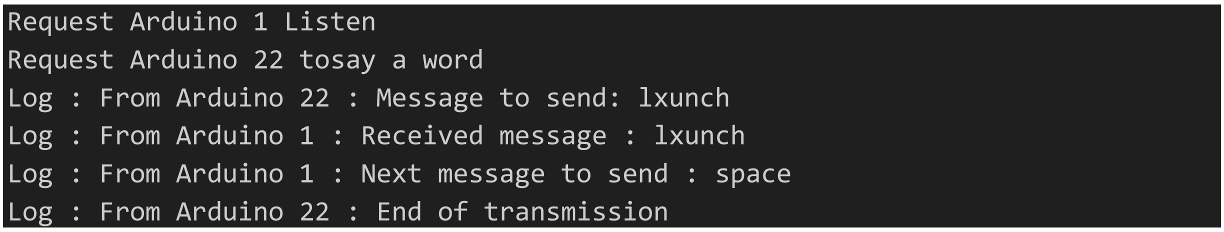
```
Log : From Arduino 22 : Received message : launch
Log : From Arduino 22 : Next message to send : space
```

- **Test 1.3:** When the Atmega 2 sends a word with no mistakes (“space”) , Atmega 1 must receive it and translate it correctly



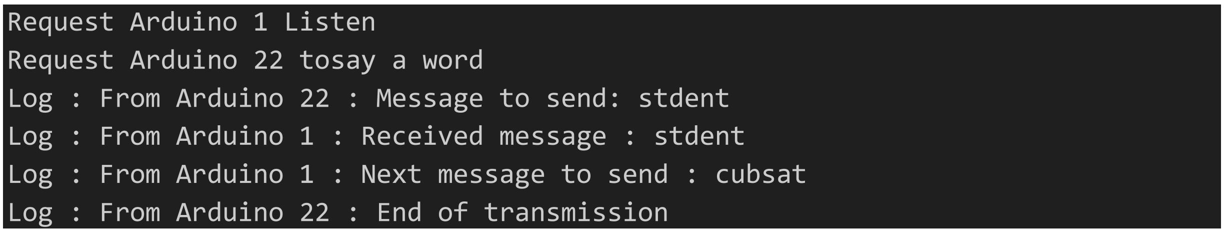
```
Request Arduino 1 Listen
Request Arduino 22 tosay a word
Log : From Arduino 22 : Message to send: space
Log : From Arduino 1 : Received message : space
Log : From Arduino 1 : Next message to send : challen&
Log : From Arduino 1 : ge
Log : From Arduino 22 : End of transmission
```

- **Test 1.4:** When the Atmega 2 sends a word with one mistake (“lxunch”), Atmega 1 must receive it and translate it correctly



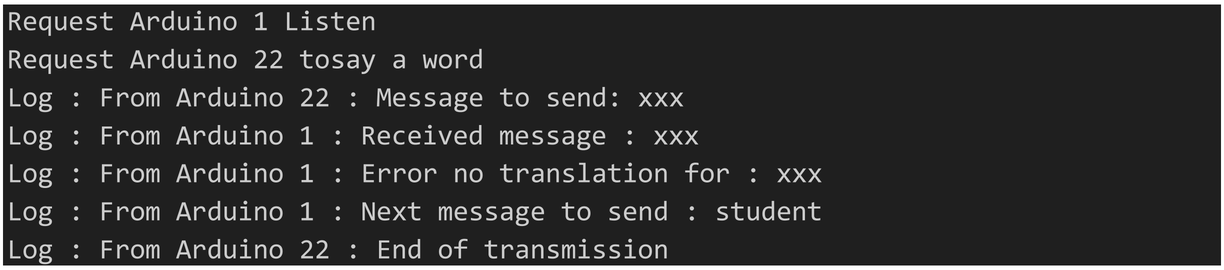
```
Request Arduino 1 Listen
Request Arduino 22 tosay a word
Log : From Arduino 22 : Message to send: lxunch
Log : From Arduino 1 : Received message : lxunch
Log : From Arduino 1 : Next message to send : space
Log : From Arduino 22 : End of transmission
```

- **Test 1.5:** When the Atmega 2 sends a word with one missing letter (“stdent”), Atmega 1 must receive it and translate it correctly.



```
Request Arduino 1 Listen
Request Arduino 22 tosay a word
Log : From Arduino 22 : Message to send: stdent
Log : From Arduino 1 : Received message : stdent
Log : From Arduino 1 : Next message to send : cubsat
Log : From Arduino 22 : End of transmission
```

- **Test 1.6:** When the Atmega 2 sends an unrecognized word (“xxx”), Atmega 1 must receive it, raise an error and find a new word to send next.



```
Request Arduino 1 Listen
Request Arduino 22 tosay a word
Log : From Arduino 22 : Message to send: xxx
Log : From Arduino 1 : Received message : xxx
Log : From Arduino 1 : Error no translation for : xxx
Log : From Arduino 1 : Next message to send : student
Log : From Arduino 22 : End of transmission
```

- **Test 1.7:** When the Atmega 2 sends an unrecognized character (“???”) Atmega 1 must receive it, raise an error and find a new word to send next.

```
Request Arduino 1 Listen
Request Arduino 22 tosay a word
Log : From Arduino 22 : Message to send: ???
Log : From Arduino 1 : Received message : ???
Log : From Arduino 1 : Error no translation for : ???
Log : From Arduino 1 : Next message to send : launch
Log : From Arduino 22 : End of transmission
```

<u>During Round 2 :</u>

- **<u>Test 1.7:</u>** The Atmega 1 must calibrate its photoresistor at the beginning of the round

```
-   ############################################
-   beginning of the round 2 :
-   No reception acceptance test
-   ############################################
-   Request Arduino 1 Init LDR
-   Request Arduino 22 Init LDR
-   Log : From Arduino 1 : Calibrating the LDR ...
-   Log : From Arduino 22 : Calibrating the LDR ...
-   Log : From Arduino 1 : LDR Calibration Done.
-   Log : From Arduino 1 : Max Luminosity = 1
-   Log : From Arduino 1 : Mean Luminosity = 0.25
-   Log : From Arduino 1 : Threshold Value = 50
-   Log : From Arduino 22 : LDR Calibration Done.
-   Log : From Arduino 22 : Max Luminosity = 7
-   Log : From Arduino 22 : Mean Luminosity = 6.35
-   Log : From Arduino 22 : Threshold Value = 56
```

- **<u>Test 1.8:</u>** When Atmega 1 doesn’t receive anything in Listen mode, it must raise an error and the Atmega1 must find a new word to send next.

```
-   Request Arduino 1 Listen
-   Request Arduino 22 to not say anything
-   Log : From Arduino 1 : Error no message received
-   Log : From Arduino 1 : Next message to send : launch
```

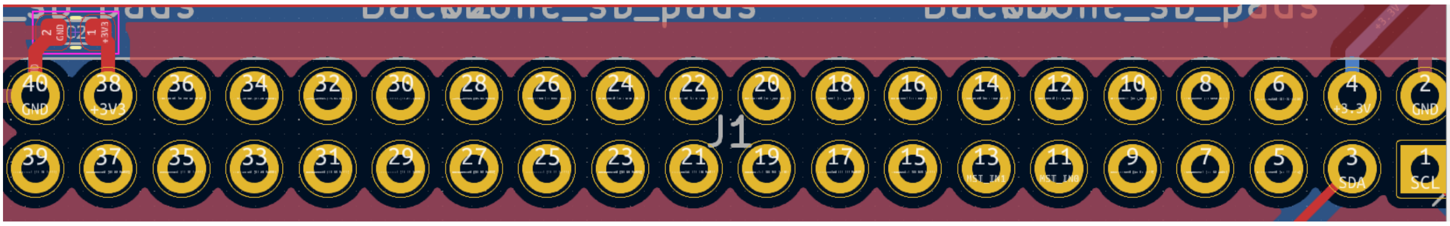
Annexe

Checklist

Test	Description	Pass	Fail
Test 1	**Test 1 is only a test of the code, checking if the basic behaviour of is correct and if it can handle small issues.**		
Test 1.1	• The code can send a log to the master. • The code can receive the InitLDR request from the master. • The code is calibrating the LDR when requested		
Test 1.2	• The code can be the first to say a word. • The code handle morse transmission correctly.		
Test 1.3 *	• The code can translate words with a one-character mistake.		
Test 1.4 *	• The code can translate words with one character missing		
Test 1.5	• The software can receive an unknown word and continue properly. • When receiving an unknown word, the code raises an error.		
Test 1.6	• The code can receive unknown characters and continue properly. • When receiving unknown characters, the code raises an error.		
Test 1.7	• The code is calibrating the LDR when requested even after some time		
Test 1.8	• The code can receive nothing during the listen mode and continue properly. • When receiving nothing during the listen mode, the code raises an error.		

The tests with * (Test 1.3 & 1.4) are not necessary to work but are good to include for completeness.

Connection



Testing Board	**Arduino**
Pin 40	Gnd
Pin 38	3V3
Pin 3	A2
Pin 1	A3