

# Exercise 6: Text Messages

## The LiFi-project

Prof. Dr. Nicolas Meseth

### OVERVIEW

In the sixth exercise of the LiFi-project you'll utilize the prototype to communicate a short text message over LiFi. This requires you to apply your combined knowledge of transmitting data over light, the binary system, and code systems.

Here are the relevant lessons of the online script. Make sure you study them carefully in order to solve this exercise:

- [Coding Text](#)
- [Logic and Arithmetic](#)
- [Binary Addition](#)

Good Luck!

### YOUR TASK

To pass this exercise, you must complete the following task and submit your results via [ILIAS](#). You find the details on the form of submission below.

### ASCII Audit Trails

One of the anticipated tasks for LiFi-equipped robots is to collect soil samples, perform analysis on those samples, and then report the results to a supervising robot. However, due to the large size of the fields in which these robots operate and the limited range of LiFi technology, the company has devised a communication system using a chain of robots. In this system, each robot will pass the information to the nearest robot until it reaches the supervising robot, who will then gather and store all the data. For this to work, the robots must be able to exchange text-based messages among each other using LiFi technology.

Write a program that sends encodes a short text message as binary numbers and sends it over the LED. The same program should be able to receive an incoming message and decode the binary numbers back to ASCII characters.

Here are the steps your program should perform. Build upon your solution from the previous exercise where possible:

1. Prompt the user to enter a short message containing only ASCII characters.
2. Verify that the input consists of ASCII characters exclusively. If the check fails, reject the input and ask the user to provide another message.
3. Convert the ASCII characters into their binary representation.
4. Transmit the resulting bytes using the LED and a three-color protocol.
5. Use the color sensor to receive the signals from the LED and decode them back into bits.
6. Convert the bits into the original text message using the ASCII code system.
7. Print the received message on the OLED display.
8. Prompt the user to enter another message for transmission. Repeat this process until the user inputs “exit”.

**Hint:** You can find an ASCII-table in PDF-format here: [ASCII Conversion Chart](#)

## SUBMISSION

For this exercise, please submit:

- `ascii_audit_trails.py`

Submit the file via the corresponding exercise in [ILIAS](#). To get started, please use the provided submission template from the LiFi-code GitHub repository: [ascci\\_audit\\_trails.py](#).

## MORE EXERCISES TO PRACTICE

The following tasks are for you to practice your skills. They are optional and not part of the submission.

### 1. To Lowercase

Write your custom function `to_lower` to convert any given string to lowercase letters only. According to [the lesson in the online script](#), the function needs to perform the following steps:

1. The function takes a string argument and iterates through each character in the string.
2. For each character, the function checks whether it is an uppercase letter by comparing its ASCII code to the range between 65 and 90, which are the ASCII codes for uppercase letters. We can get the ASCII code of a character using the `ord()` function.
3. If the character is an uppercase letter, the function adds 32 to its ASCII code, which effectively converts it to the corresponding lowercase letter. Then, it converts the new ASCII code back to a character using the `chr()` function and adds it to a result string variable.
4. If the character is not an uppercase letter, the function simply adds the character to the result variable without modification.
5. After all characters have been processed, the function returns the result variable containing the converted string.