Exercise 3: On And Off

The LiFi-project

Prof. Dr. Nicolas Meseth

OVERVIEW

In the third exercise of the LiFi-project, you're going to enhance your understanding of the binary system and practice your coding skills at the same time.

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

- 7 - Binary Numbers
- 8 - The Rotary Encoder
- 9 - Collections

Good Luck!

YOUR TASKS

To pass this exercise, you must complete the following tasks and submit your results via ILIAS. You find the details on the form of submission below.

1. Binary Dolmetcher

The LiFi prototype will ultimately send data in the form of bits and bytes over light signals. In this exercise, you take one step in that direction:

- Write a function with the name `decimal_to_binary`, which takes a whole decimal number and converts it into a string of zeroes and ones that represents the number in the binary system.
- Write another function `binary_to_decimal`, which does the opposite: it takes a binary number as a string of zeroes and ones, e.g. `1101`, and converts it into its equivalent in the decimal system (here: `13`).

Note: For the conversion from binary to decimal and vice versa, you may not use built-in functions like `bin()` or `int()`, or any external module that performs this particular task for you. The converter must be written "by hand" using the concepts we introduced in the previous lessons. For tasks such as converting a string to a number, you can still use `bin()`.

2. Count Control

To test the LiFi prototype in a live environment and with no laptop at hand, a future user will have to input data using the rotary encoder. This exercise trains your skills with this particular device.

- Write a program that counts in binary. When the user turns the rotary encoder's knob, the count c is increased (if turned right) or decreased (if turned left) by a number n that can be either 1, 2, 4, 8 or 16.
- The count c cannot be less than 0, and if $c = 0$ and the user keeps turning left, c stays at 0.
- Initially, $n = 1$, and every time the user pressed the knob, n is increased to the next larger number in the list. When $n = 16$ and the user presses the knob, it starts again at $n = 1$.
- With every turn of the knob, the new value for c is printed to the console as a **binary number**. Also, if the user presses the knob, the user gets informed through a message on the console, such as *"The value for* n is now 8!", where the value for n is shown in decimal.

SUBMISSION

For this exercise, please submit:

- Two Python programs, each in a separate file for the tasks above:
  - `binary_dolmetcher.py`
  - `count_control.py`

Submit all files via the corresponding exercise in ILIAS.

MORE EXERCISES TO PRACTICE

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

3. Just The Right Light

In this exercise, you combine the two hardware devices, the LED and the rotary encoder in a single program. Additionally, you apply your knowledge of the RGB code.

- Write a program that allows the user to control the color of the LED using the rotary encoder. The user can increase or decrease the intensity of the currently controlled base color by turning the knob in the right or left direction.
- To change the currently controlled base color, the user must press the rotary encoder's knob. With each press, the controlled color changes from red to green, and then to blue. The cycle starts over again.
- The intensity for any of the three base colors cannot be lower than 0 or exceed 255. When the intensity is 255 and the user keeps turning right, the intensity rewinds to 0. The same is true for the other direction, where the intensity jumps to 255 when the user keeps turning left at 0.

4. Fast Reaction Game

We can use the LED and the rotary encoder to build a small game. The game tests a player's reaction time and faster players score more points. The game works as follows:

- When the program starts, prompt the user to hit ENTER to start a new game.
- Now, let the LED change randomly between the colors red, green, blue, and yellow. A color is shown for a period p, which is initially set to 1 second. Let p slowly decrease with the duration of a game.
- When the color green shows up, the player should press the rotary encoder's knob as fast as possible. The time r it took the player to hit the knob is negatively correlated with the points she gets. If it takes longer than a certain time r_{max}, the player receives 0 points for that round. Print the achieved points as well as the total point to the console after every round.
- Pressing the knob at a different color than green results in a penalty. The player is also penalized if she fails to press the know during a green color phase.
- Show random colors until the color green showed up 10 times (ten rounds). After every time the LED was green, the period p decreases by a small amount.

- When the game finishes, print the total points to the console. Optionally, ask for the player's name and keep the score in a high score table.

MORE QUESTIONS TO PRACTICE

Try to answer the following questions to practice your understanding of the topics around the LiFi project. The questions are optional and not part of the submission.

1. Explain why the bit is the smallest possible unit of information!
2. Discuss why computers use binary rather than decimal or other number systems. What are the advantages and disadvantages?
3. How many bits are needed to represent the following numbers:

   a) 64 different values
   b) 128 different values
   c) 1024 different values

4. Convert the following decimal numbers to binary:

   a) 12
   b) 27
   c) 85
   d) 128

5. Convert the following binary numbers to decimal:

   a) 1001
   b) 1110
   c) 101011
   d) 1101010

6. In a hexadecimal (base-16) system, the numbers 0-9 are not enough to represent all the possible values. How is this problem solved?
7. The number 255 in base-10 requires 8 bits for its binary representation. This makes the binary number 5 digits longer than the same number in the decimal system. What is the fundamental relationship between the number of available symbols and the length of a message (in this case, a number)?
8. If you buy a computer today, you will likely find that it has a 64-bit architecture. A few years ago, the standard for most computers was a 32-bit architecture, while devices like the Commodore 64 or the Nintendo NES were based on an 8-bit architecture:

   a) Research on the internet what this specification means!

b) What does the transition from 32-bit to 64-bit architecture mean in practical terms?

9. What is the difference between a kilobyte and a kibibyte?