Lab Assignment 1 CS301 2022FALL

You are a spy on a special mission. On this mission, you will need to control the flashing of the street lamp to send messages to the contact, and the information is encode by the famous **Morse code**.

Morse code is a method used in telecommunication to encode text characters as standardized sequences of two different signal durations, called **dots** `.` and **dashes** `-`. The coding rules are shown below:

Character	Code	Character	Code	Character	Code	Character	Code	Character	Code
A	`.-`	B	`-...`	C	`-.-.`	D	`-..`	E	`.`
F	`..-.`	G	`--.`	H	`....`	I	`..`	J	`.---`
K	`-.-`	L	`.-..`	M	`--`	N	`-.`	O	`---`
P	`.--.`	Q	`--.-`	R	`.-.`	S	`...`	T	`-`
U	`..-`	V	`...-`	W	`.--`	X	`-..-`	Y	`-.--`
Z	`--..`								

Part 1 (80%)

In this part, you need to use the **UART** in non-blocking mode to realize the information transfer with the STM32 Development Board.

1. Processing single alphabet Morse Code (20%):
   - When receiving the **dots** `.` from PC , blink LED0 one time;
   - When receiving the **dashes** `-` from PC , blink LED1 one time;
   - When receiving successive **dots** and **dashes** from PC, LED0 and LED1 should blink in order. For example, when receiving the message `.--.`, blink LED0 one time, then blink LED1 two times, finally blink LED0 one time. (Note that the LED0 and LED1 cannot flash at the same time when receiving successive **dots** and **dashes**)

2. Processing multiple alphabet Morse Code (20%):
   - We use character `/` to represent the end of alphabet Morse Code when there are multiple alphabet Morse Codes. When receiving `/` from PC, blink LED0 and LED1 two times at the same time.
   - For example, the Morse Code of `HELLO` is `..../././.-../.-../---/`.

3. Decode and encode Morse Code (40%):
   - Decode: After receiving the Morse Code from PC, send the **decoded characters** to PC. Then LED0 and LED1 should blink in the order of Morse Code.

    - Encode: After receiving the alphabet to be translated from PC, send the **encoded Morse Code** to PC. Then LED0 and LED1 should blink in the order of Morse Code
    - Design mechanism to switch between Encode and Decode states. For example, one case is that after receiving the message `1` (or `0`) from PC, it becomes Encode (or Decode) state; Another case is that you can prefix the message with `E:` (`D:`) which stands for encode or decode.

Part 2 (20%)

In this section, you can design own requirements that you think are reasonable, and we will score them according to the level of creativity and difficulty. The following are some examples:

- Exception handling (10%)
    - Decode: Abnormal input like `?,a,1` or invalid Morse code like `.---.`. Send error message and corresponding error Moss Code to PC, for example `Decode Error: .---.`. Then **turn on LED0** and **blink LED1 three times** at same time. After that, continue to process the next alphabet Morse Code.
    - Encode: Non-alphabetic character input like `1, ,`. Send error message and corresponding error non-alphabetic character to PC, for example `Encode Error: 1`. Then **turn on LED1** and **blink LED0 three times** at same time. After that, continue to process the next character to be translated.

- Two LED process Morse code in parallel (10%)
    1. Design a way for a LED to represent **dots** `.` and **dashes** `-`.
        - When receiving the **dots** `.` from PC , blink LED0 one time;
        - When receiving the **dashes** `-` from PC , blink LED0 two times;
        - Use long extinguish (long silence) to separate **dots** and **dashes**
        - No need to consider multiple alphabet Morse Code, that is, no need to consider the symbol `/`
    2. The transmitted information needs to specify which LED is to be processed. For example, when receiving the message `LED0: .--.`, the led action unfolds in the following order: blink LED0 one time, long silence, blink LED0 two times, long silence, blink LED0 two times, long silence, blink LED0 one times.
    3. Two LED process Morse code in parallel. When LED0 is blinking, you can specify LED1 to handle the Moss code. To demonstrate the parallel process, you can set the long silence time longer.

- Show decode and encode progress (5%):
    1. Decode: Send the corresponding decoded alphabet before the light blinks according to the corresponding Morse code. For example, when decode `..../././.-../.-../---/`, first send the message `%time% H` to PC, and blink the led according Morse Code `..../`, then send the message `%time% E` to PC, and blink the led according Morse Code `./` and so on. `%time%` is the current time.

2. Encode: Send the corresponding encrypted Morse Code before the light blinks according to the corresponding Letter to be translated. For example, when decode `HELLO`, first send the message `%time% ....` to PC, and blink the led according Morse Code `..../`, then send the message `%time% .` to PC, and blink the led according Morse Code `./` and so on. `%time%` is the current time.

- Use the keys to pause or continue the encode or decode process, and send the message `%time% pause` or `%time% continue` to PC (5%)
  - Note that after pause the encode or decode process, the data transferred by UART needs to be stored in the cache, and when the process continues, it needs to be processed in order.

- When receiving alphabets, use keys to send Morse Code, for example use key0 as **dots** `.`, key1 as **dashes** `-` and key_wakeup as `/`. (10%)
  - Realize button anti-shake function.
  - You need to record the information of keys0 and key1. When `/` is received, that is key_wakeup is pressed, the led light starts flashing.
  - ~~Note that exception messages need to be handled.~~

Submission demands

1. Finish the assignment before DDL.
2. Package the whole project into a compressed package and submit on Sakai site.