Lab Assignment 2 CS301 2023FALL

Suppose your younger brother in elementary school is learning addition and subtraction within 10, but he feels that he's not proficient enough. Knowing that you are good at programming, he wants you to design game to help him practice addition and subtraction.

Part 1 Basic Part (80%)

During this part, you need yo use **UART** in non-blocking mode for communication between the STM32 Development Board and the PC. On the PC side, you can use a serial debugging assistant, with the keyboard as the tool for inputting answers. Additionally, you will use other features such as **IWDG** and **timers**. The main function is to send random (or predefined) equations to the PC , such as `1+2=?` , and the user needs to using the keyboard to input the right answer in a limited time.

1. Send data and receive data . (40%)
    - In the process of sending data, we assume that it is an addition exercise . Send `1+2=?` to the PC. The user need to input the right answer and move on to the next equation . (20%)

      So you should compare the user input and the right answer.

        - If the user input the right answer in time correctly , **blink the green LED** one time and send `[INFO] True! Do a good job!`
        - If the user input the wrong answer , **blink the red LED** one time and send `[INFO] Wrong! Please try again!`
    - Set time **1 min** to complete the exercise . In a complete exercise, the user need pass at least **5** equations (for both addition and subtraction) . (20%)
        - Transmit the remaining time `[Warning] ? second left!` to PC when there are only 10 .. or 3,2,1 seconds left.
        - If the user can pass the exercise on time , send `[INFO] You pass the practice! Please press KEY0 to try again .` . Then use software to reset .

          Hint : the reset function is `HAL_NVIC_SystemReset()`.
2. Program mechanism . (40%)
    - If the user can't pass the exercise on time , the program should be reset automatically by **IWDG** , and send `[INFO]Time out. Game over. Please press KEY0 to try again .` (20%)
    - when the program is reset , the user should be able to start the game by pressing `KEY0`. (20%)

Part 2 Advance Part (20%)

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

- Switch the addition or subtraction (10%) :
    - Select addition exercise or subtraction exercise by pressing `KEY1`.
    - Send `[UPDATE] This is the addition(subtraction) exercise` to PC.

- The equation can be randomly generated (5%) :
  - Can randomly generate numbers and add or subtract equations. Such as the first equation is `1+3=?` , the next equation is `5-4=?` .
  - Add equation limit the result no more than 10 , subtract equation limit the result cannot be negative.
- Extended computation (15%) :
  - Expand addition and subtraction within 10 to addition, subtraction, multiplication and division within 100 .Such as the first equation is `12+31=?` , the next equation is `5*4=?` .
  - The dividend of a division formula is non-zero and the result is an integer .
- Switch time limit (10%) :
  - Change the time limit by `KEY_WK` during the program is running.
  - Send `[UPDATE] Time limit changed: ? min.` to PC.
- Punishment mechanism (10%) :
  - If the user input the wrong answer , the user need to answer more questions correctly to pass the exercise. For example, if you get a wrong answer once, you need to answer 6 equations to pass the test.

Help

You may need help as only some of you are skilled C programmers.

- Data type conversion
  - char convert to int (The number ranges from 0 to 9) .

```
char c = '0';
int i = c - '0';// 0
```

  - int convert to char (The number ranges from 0 to 9) .

```
int i = 5;
char c4 = i + '0'; // 5
```

- Here are a few useful tools in the c library `string.h` :
  - include it

```
#include <string.h>
```

  - define a string

```
char string[10];
char *str1 = "abcdefghi";
```

  - copy

```
char *strcpy(char *destin, char *source);
char *strncpy(char *dest, char *src, size_t n);
```

  - concatenate

```
 char *strcat(char *destin, char *source);
```

- compare

```
int strcmp(char *str1, char *str2);
```

- length

```
size_t strlen(const char *s);
```

And more...

Search it on the Internet may help you more. If you don't like new stuffs, just use for loops and treat the string as simple `char` arrays.

Submission demands

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