Lab Assignment 2

Mohit Maurya (22110145)

Parag Sarvoday Sahu (22110179)

Q1: First names in Morse code

C Code:

```
  /* USER CODE BEGIN 1 */

 const char *parag[] = {".", "-", "-", ".", "|", ".", "-",
"|", ".", "-", ".", "|", ".", "-", "|", "-", "-", ".",
 ",", "-", "-", "|", "-", "-", "-", "|", ".", ".", ".", ".",
"|", ".",".", "|", "-"};
 int time_unit = 500;
 int size_parag = sizeof(parag) / sizeof(parag[0]);

  /* USER CODE END 1 */
```

We define an array containing the required Morse code characters corresponding to our first names i.e. Parag and Mohit (see figure below). We use "," as a delimiter to separate the symbols for the two first names.

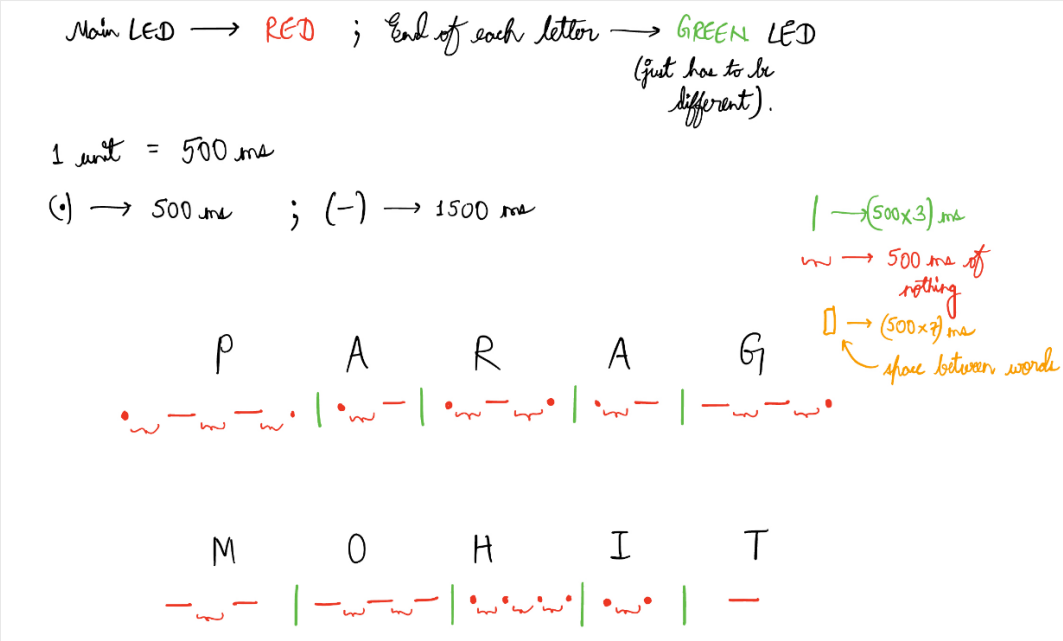


Fig 1. Finding out the Morse code characters corresponding to our names

We also find the size of the array to iterate through its elements at the later stage of the program.

```
/* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */


    /* USER CODE BEGIN 3 */
   for (int i=0; i < size_parag; i++){
              if (parag[i] == ","){
                   HAL_Delay(6 * time_unit);
              }
              if (parag[i] == "|"){
                   HAL_GPIO_WritePin(GPIOB, LD1_Pin, 1);
                   HAL_Delay(3*time_unit);
                   HAL_GPIO_WritePin(GPIOB, LD1_Pin, 0);
              }
              else{
                   if (i > 0 & parag[i-1] != "|"){
                        HAL_Delay(time_unit);
                   }

                   if (parag[i] == "."){
                      HAL_GPIO_WritePin(GPIOB, LD3_Pin, 1);
                      HAL_Delay(time_unit);
                      HAL_GPIO_WritePin(GPIOB, LD3_Pin, 0);

                              }
                   if(parag[i] == "-"){
                        HAL_GPIO_WritePin(GPIOB, LD3_Pin, 1);
                        HAL_Delay(3 * time_unit);
                        HAL_GPIO_WritePin(GPIOB, LD3_Pin, 0);

                   }
              }


        }
  }
  /* USER CODE END 3 */
```

Next, we iterate through all the elements of the defined array of Morse code characters. In each iteration, we turn on the right LED and apply the appropriate amount of delay corresponding to the character.

Q2: Roll numbers in binary number system using LEDs

C Code:

```
/* USER CODE BEGIN Includes */
 int roll_no[] = {2,2,1,1,0,1,4,5,10,2,2,1,1,0,1,7,9};
 int time_unit = 1000;
 int size_roll_no = sizeof(roll_no) / sizeof(roll_no[0]);
/* USER CODE END Includes */
```

```
  /* USER CODE BEGIN WHILE */
  while (1)
  {
    /* USER CODE END WHILE */




    /* USER CODE BEGIN 3 */
   for (int i =0; i < size_roll_no; i++){
      if(roll_no[i] == 10){
                HAL_Delay(2*time_unit);
           }
      else {
           if (i>0){
           HAL_Delay(3*time_unit);
           }

           if(roll_no[i] == 0){
                HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, 1);
                HAL_Delay(time_unit);
                HAL_GPIO_WritePin(GPIOB, GPIO_PIN_10, 0);
                     }
                int shift_num = roll_no[i] ;
      HAL_GPIO_WritePin(GPIOB, LD1_Pin, shift_num & 1 );
                shift_num = shift_num >> 1;
      HAL_GPIO_WritePin(GPIOB, LD2_Pin, shift_num & 1 );
                shift_num = shift_num >> 1;
      HAL_GPIO_WritePin(GPIOB, LD3_Pin, shift_num & 1 );
                shift_num = shift_num >> 1;
      HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, shift_num & 1 );

                HAL_Delay(time_unit);

                HAL_GPIO_WritePin(GPIOB, LD1_Pin, 0 );
                HAL_GPIO_WritePin(GPIOB, LD2_Pin, 0 );
```

```
                HAL_GPIO_WritePin(GPIOB, LD3_Pin, 0 );
                HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, 0);

                }
            }
 HAL_Delay(10*time_unit); //to indicate program restart
}

/* USER CODE END 3 */
```

**Pins Used:**
1) LD1 (green) LSB

2) Pin PB15 as MSB

3) Pin PB10 as 0 showing LED

We have created a list of numbers (named *roll_no*) that contains the roll numbers of each member. Each digit in a roll number is comma-separated, and the roll number of each member is separated by a random delimiter, which is the number 10 in this case.

In the *while* loop, we use a *for* loop to iterate through the list. If the number 10 is detected, a delay of *2 × time_unit* is introduced (where *time_unit* represents a unit delay of 1 second). According to the question, this delay should be 5 seconds, but here, it is set to 2 seconds, as the remaining 3 seconds are compensated for below. Otherwise, an intentional delay of 3 seconds is added, except at the start of the list, where no delay should be present before the first digit. However, this delay will be applied after each digit is displayed.

To represent the binary equivalent of the digit at index *i*, we right-shift the digit by 1 and perform a bitwise AND operation with 1 to extract the least significant bit (LSB). This LSB is then used to either set or reset the LED. This process is repeated four times (for 4 bits) to obtain the binary representation of the digit.

Example:

2 is represented by 0010
Let variable be *shift_num* 2.

Now:
LSB = *shift_num* & 1 => 0  This will reset the LED LD1 (LSB)
*shift_num* = *shift_num* >> 1
LSB = *shift_num* & 1 => 1  This will set the LED LD2
*shift_num* = *shift_num* >> 1

LSB = *shift_num* & 1 => 0 This will reset the LED LD3
*shift_num* = *shift_num* >> 1
LSB = *shift_num* & 1 => 0 This will reset the LED Pin PB15 (MSB)