ASSIGNMENT NOTE

Subject: Embedded Programming Principle

Topic: Pass Code

Student: Minh Quan Tran

July 7th, 2024

Table of Contents

TABLE OF FIGURES

1. Objectives

- Build a circuit that check passcode.
- Write a C program in STM32 CUBE IDE that check passcode from Putty.
- Flash code and debug project.

2. Requirement

2.1 Requirement

Write a program that will do the following:

- accept up to a 4 digit value from the virtual serial port on the STM Board.
- If the 4 digit value is one of the 10 pass-codes stored in an array, issue an auditory feedback signal and the text "access granted" to the serial port.
- If it is not one of the 10 allowable pass-codes, issue another appropriate auditory feedback signal and the text "access denied".
- Also illuminate a red led if the access is not granted and a green led if access is granted.

As always follow the coding standards for the ESD department.

3. Hardware and Software design

3.1 Hardware Design

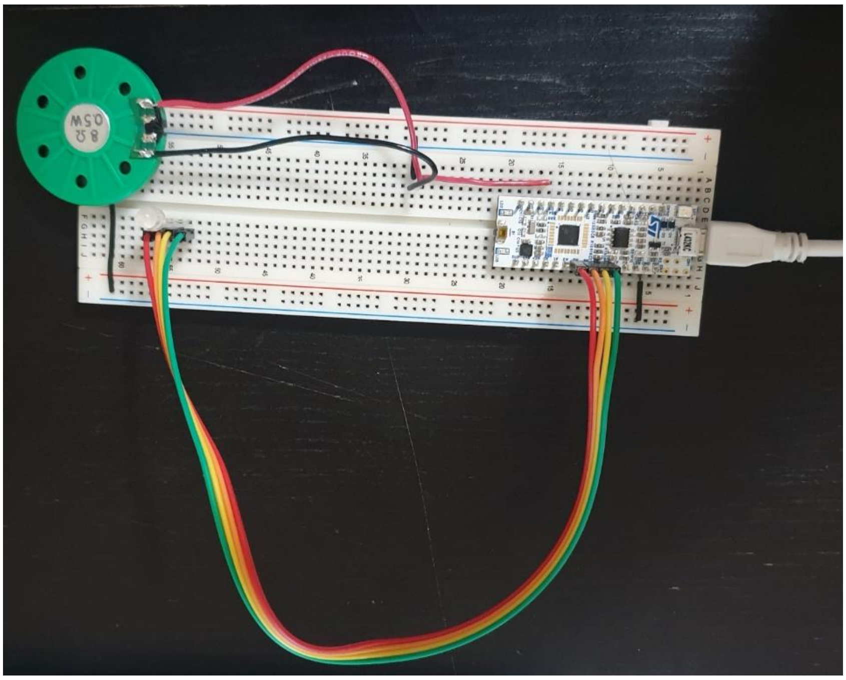
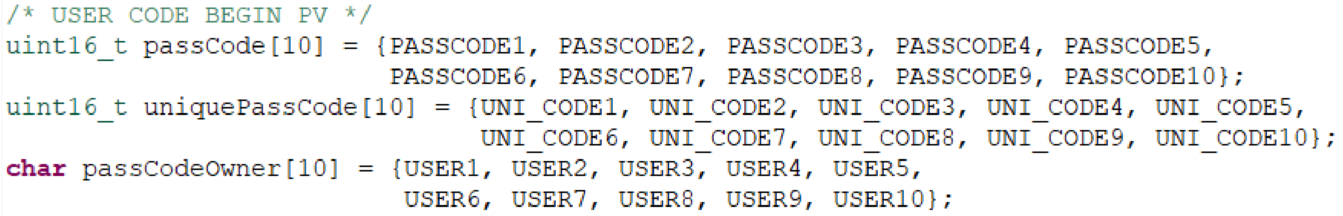


Figure 3-1: Breadboard's circuit

3.2 Software Design

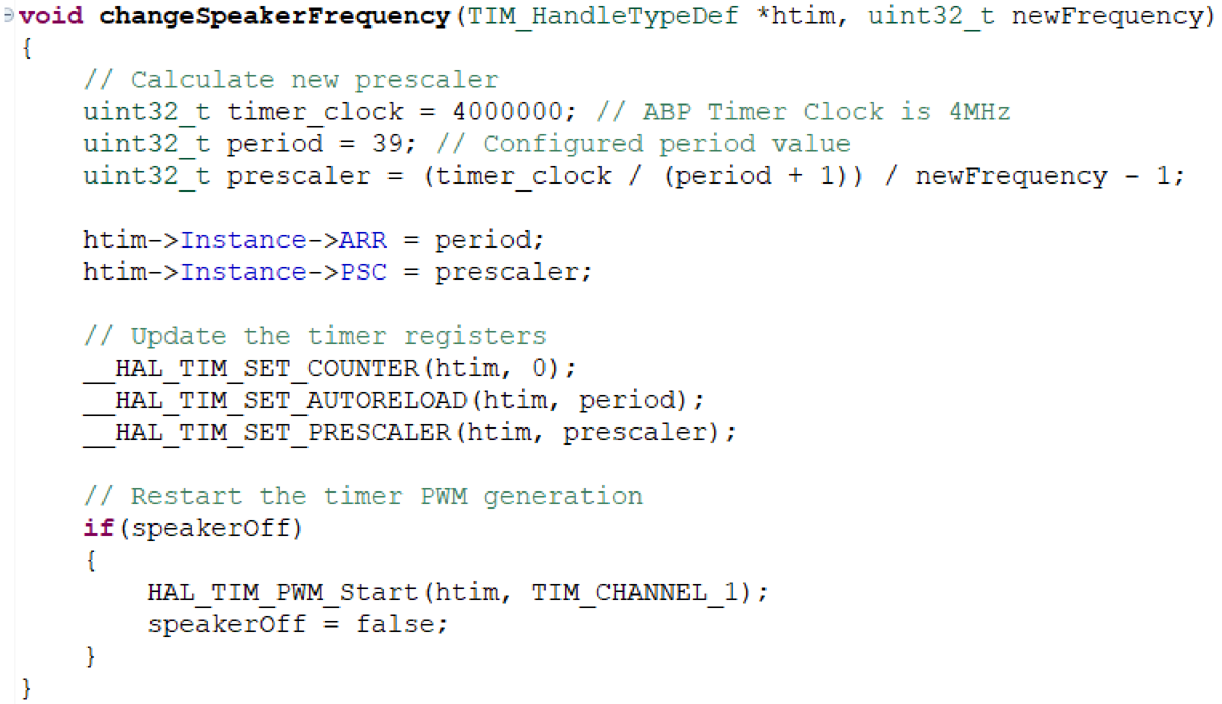
Create an array Passcode [10], PasscodeOwner[10] and UniquePassCode that will contain all 10 passcode, their user and their own Unique passcode.



```
/* USER CODE BEGIN PV */
uint16_t passCode[10] = {PASSCODE1, PASSCODE2, PASSCODE3, PASSCODE4, PASSCODE5,
                         PASSCODE6, PASSCODE7, PASSCODE8, PASSCODE9, PASSCODE10};
uint16_t uniquePassCode[10] = {UNI_CODE1, UNI_CODE2, UNI_CODE3, UNI_CODE4, UNI_CODE5,
                               UNI_CODE6, UNI_CODE7, UNI_CODE8, UNI_CODE9, UNI_CODE10};
char passCodeOwner[10] = {USER1, USER2, USER3, USER4, USER5,
                          USER6, USER7, USER8, USER9, USER10};
```

Figure 3-2: Declaring PassCode

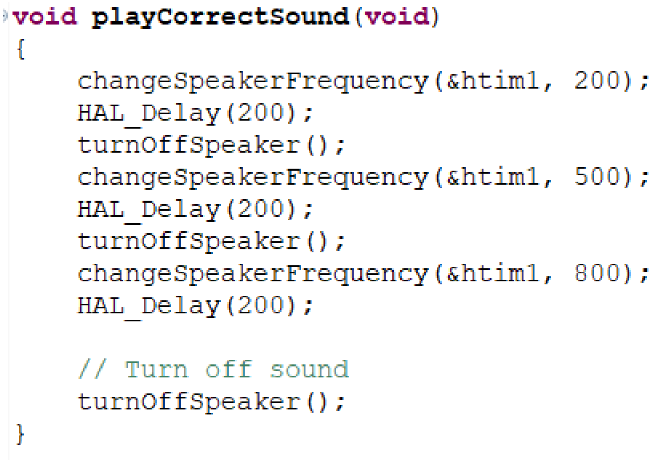
Create a simple changeSpeakerFrequency function that will calculate the new prescaler whenever change to a new frequency.



```
void changeSpeakerFrequency(TIM_HandleTypeDef *htim, uint32_t newFrequency)
{
    // Calculate new prescaler
    uint32_t timer_clock = 4000000; // ABP Timer Clock is 4MHz
    uint32_t period = 39; // Configured period value
    uint32_t prescaler = (timer_clock / (period + 1)) / newFrequency - 1;

    htim->Instance->ARR = period;
    htim->Instance->PSC = prescaler;

    // Update the timer registers
    __HAL_TIM_SET_COUNTER(htim, 0);
    __HAL_TIM_SET_AUTORELOAD(htim, period);
    __HAL_TIM_SET_PRESCALER(htim, prescaler);

    // Restart the timer PWM generation
    if(speakerOff)
    {
        HAL_TIM_PWM_Start(htim, TIM_CHANNEL_1);
        speakerOff = false;
    }
}
```

Figure 3-3: changeSpeakerFrequency's function

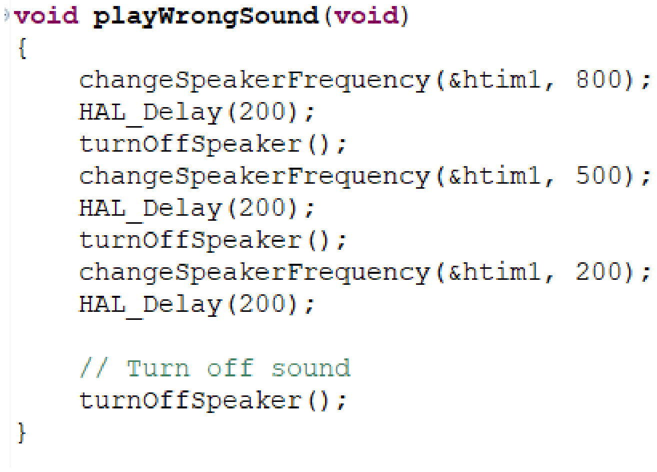
It also require to have an auditory feedback so create 2 more functions, one for when the correct passcode was typed and one for the wrong one.



```
void playCorrectSound(void)
{
    changeSpeakerFrequency(&htim1, 200);
    HAL_Delay(200);
    turnOffSpeaker();
    changeSpeakerFrequency(&htim1, 500);
    HAL_Delay(200);
    turnOffSpeaker();
    changeSpeakerFrequency(&htim1, 800);
    HAL_Delay(200);

    // Turn off sound
    turnOffSpeaker();
}
```

Figure 3-4: playCorrectSound’s function

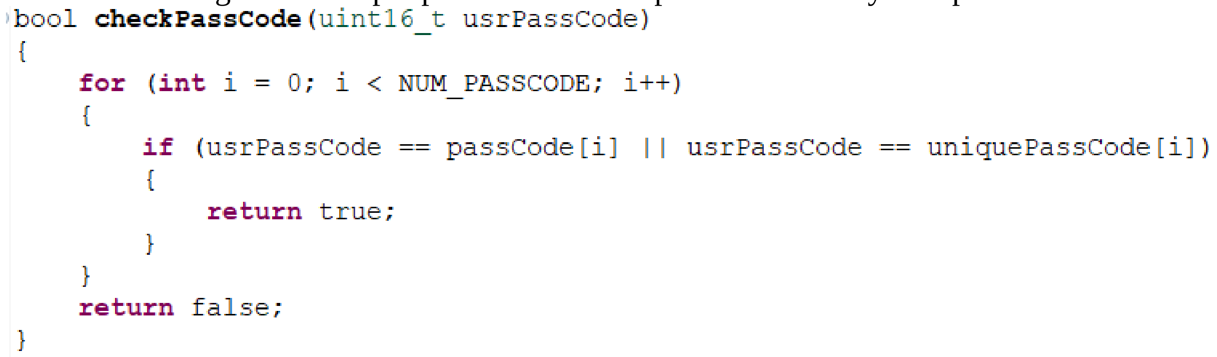
This function will create a ring tone from 3 different frequency.



```
void playWrongSound(void)
{
    changeSpeakerFrequency(&htim1, 800);
    HAL_Delay(200);
    turnOffSpeaker();
    changeSpeakerFrequency(&htim1, 500);
    HAL_Delay(200);
    turnOffSpeaker();
    changeSpeakerFrequency(&htim1, 200);
    HAL_Delay(200);

    // Turn off sound
    turnOffSpeaker();
}
```

Figure 3-5: playWrongSound's function

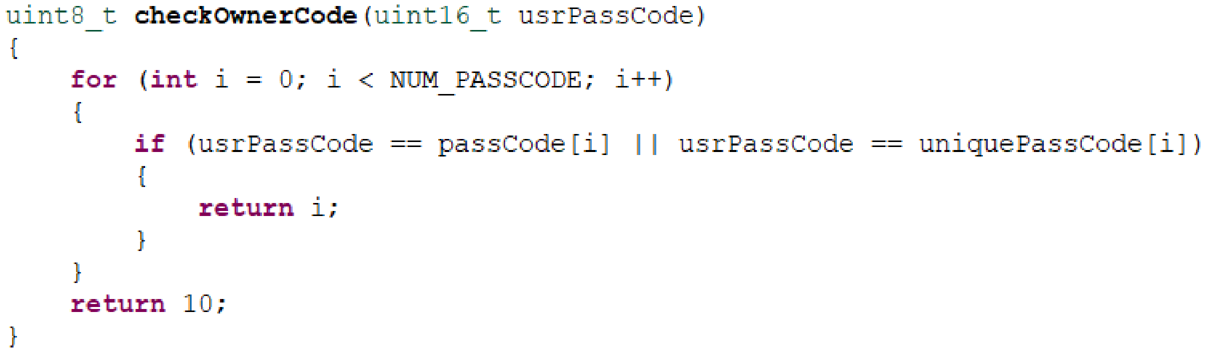
Function for checking the user input passcode and compare it with the system passcode



```
bool checkPassCode(uint16_t usrPassCode)
{
    for (int i = 0; i < NUM_PASSCODE; i++)
    {
        if (usrPassCode == passCode[i] || usrPassCode == uniquePassCode[i])
        {
            return true;
        }
    }
    return false;
}
```

Figure 3-6: checkPassCode's function

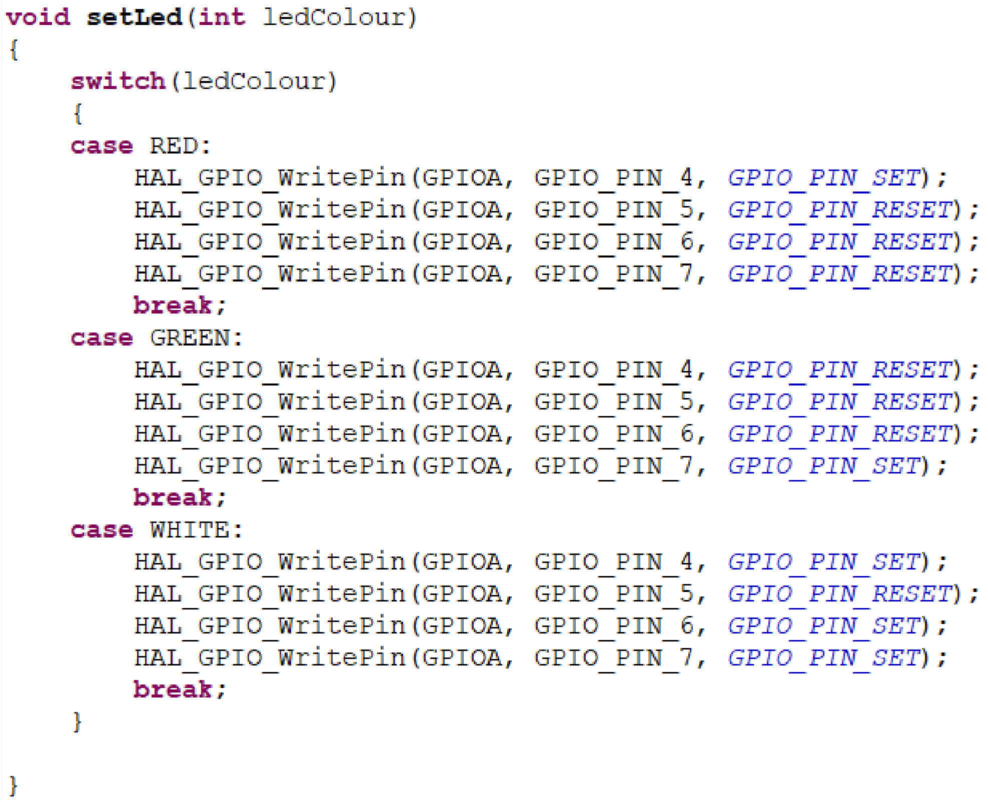
Also a function in which will find the owner of that PassCode



```
uint8_t checkOwnerCode(uint16_t usrPassCode)
{
    for (int i = 0; i < NUM_PASSCODE; i++)
    {
        if (usrPassCode == passCode[i] || usrPassCode == uniquePassCode[i])
        {
            return i;
        }
    }
    return 10;
}
```

Figure 3-7: checkOwnerCode 's function

A function to change the RGB LED color



```
void setLed(int ledColour)
{
    switch(ledColour)
    {
    case RED:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_RESET);
        break;
    case GREEN:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
        break;
    case WHITE:
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_4, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, GPIO_PIN_RESET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, GPIO_PIN_SET);
        HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, GPIO_PIN_SET);
        break;
    }

}
```

Figure 3-8: setLed's function

And a function to reset the user passcode

```
void resetPassCode(char usrReset)
{
	int usrUniqueCode, usrNewCode, ownerCodeName = 0;
	int numberAttemptLeft = 0;

	if (usrReset == 'y')
	{
		for (int i = NUM_ATTEMPT; i > 0; i--)
			{
				printf("Please type in your UNIQUE passcode: \r\n");
				scanf("%d", &usrUniqueCode);
				if (checkPassCode(usrUniqueCode) == USERCODE_MATCH)
				{
					printf("Access Granted\r\n");
					setLed(GREEN);
					playCorrectSound();
					numberAttemptLeft = 3;
					break;
				}
				else
				{
					printf("Access Denied\r\n");
					printf("You have %d attempt left \r\n", i - 1);
					setLed(RED);
					playWrongSound();
					numberAttemptLeft = i - 1;
					continue;
				}
			}

		// Zero attempt left for resetting pass code
		if (numberAttemptLeft == 0)
```

```
        {
            printf("Please contact the owner for further assist\r\n");
            printf("~~~~~~~~~~~~~~~\r\n");
            return;
        }

        // Resetting user passcode
        else
        {
            ownerCodeName = checkOwnerCode(usrUniqueCode); //Get Owner Name
            printf("Welcome Mrs %c\r\n", passCodeOwner[ownerCodeName]);
            printf("Please type in your new code\r\n");
            scanf("%d",&usrNewCode);
            passCode[ownerCodeName] = usrNewCode;
            printf("Your new passcode have been saved, please log in
again\r\n");
            printf("~~~~~~~~~~~~~~~\r\n");
            return;
        }
    }
    else
    {
        printf("~~~~~~~~~~~~~~~\r\n");
        return;
    }
}
```

4. Result

Video demo's: MTran_Assignment2_Embedded_vid.mp4

REFERENCES