LAB: USART- LED,Bluetooth

Embedded Controller Lab Report

LAB: Input Capture - Ultrasonic

Date: 2023.11.16

Author: Hanmin Kim

Github:https://github.com/ansterdaz/HKim/tree/main

Demo video:

LED: https://youtu.be/DysFVkn8NI4

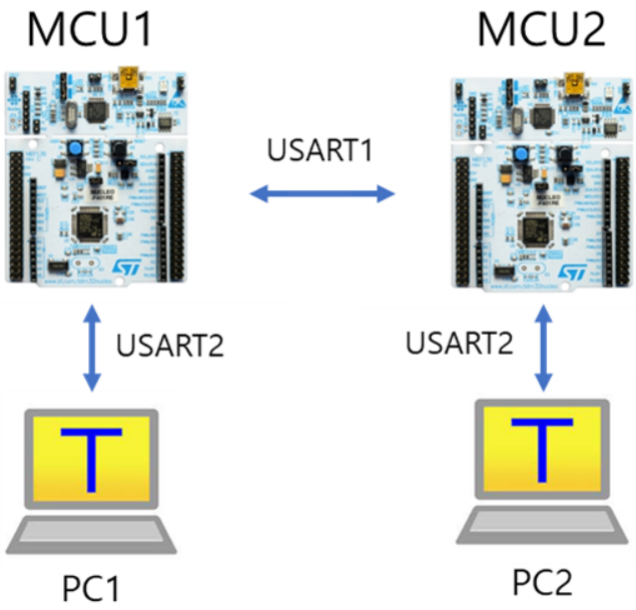
Bluetooth: https://youtu.be/9s9FBUpjwYs

Introduction

In this lab, the focus is on configuring and utilizing the 'USART (Universal Synchronous Asynchronous Receiver Transmitter)' of the MCU. The goal is to understand how to establish communication between a PC and the MCU, as well as between two MCUs, using wired serial communication.

Mission 1: Control LED(LD2) of each other MCU.

Mission 2: Run DC motors with Bluetooth



Requirement

Hardware

MCU

- NUCLEO-F411RE

Actuator/Sensor/Others:

- DC motor, DC motor driver(L9110s)

- Bluetooth Module(HC-06),

Software

Keil uVision, CMSIS, EC_HAL library

Problem 1: EC HAL library

Using HAL library

Download [ecUART_student.c ecUART_student.h]

Change the file names as

- ecUART.c, ecUART.h

ecUSART.h

```
// Configuration UART 1, 2 using default pins
void UART1_init(void);
void UART2_init(void);
void UART1_baud(uint32_t baud);
void UART2_baud(uint32_t baud);

// USART write & read
void USART1_write(uint8_t* buffer, uint32_t nBytes);
void USART2_write(uint8_t* buffer, uint32_t nBytes);
uint8_t USART1_read(void);
uint8_t USART2_read(void);

// RX Interrupt Flag USART1,2
uint32_t is_USART1_RXNE(void);
uint32_t is_USART2_RXNE(void);
```

Problem 2: Communicate MCU1-MCU2 using RS-232

Procedure

1. Create a new project under the directory `\repos\EC\LAB\LAB_USART_LED`

- The project name is "**LAB_USART_LED".**
- Create a new source files named as "**LAB_USART_LED.c"**

2. Include your updated library in `\repos\EC\lib\` to your project.

- **ecGPIO.h, ecGPIO.c**
- **ecRCC.h, ecRCC.c**
- **ecUART.h, ecUART.c**
- and other necessary header files

3. Connect each MCUs to each PC with **USART 2** via USB cable (ST-Link)

- MCU1-PC1, MCU2-PC2

4. Connect MCU1 to MCU2 with **USART 1**

- connect RX/TX pins externally as
    - MCU1_TX to MCU2_RXD
    - MCU1_RX - MCU2_TX

5. Send a message from PC_1 by typing keys on Teraterm. It should send that message from MCU_1 to MCU_2.

6. The received message by MCU_2 should be displayed on PC_2.

7. Turn other MCU's LED(LD2) On/OFF by sending text:

- "**L**" for Turn OFF
- "**H**" for Turn ON

Configuration

Type	Port - Pin	Configuration
System Clock		PLL 84MHz
USART2 : USB cable (ST-Link)		No Parity, 8-bit Data, 1-bit Stop bit, 38400 baud-rate
USART1 : MCU1 - MCU2	TXD: PA9 RXD: PA10	No Parity, 8-bit Data, 1-bit Stop bit, 38400 baud-rate
Digital Out: LD2	PA5	

Code

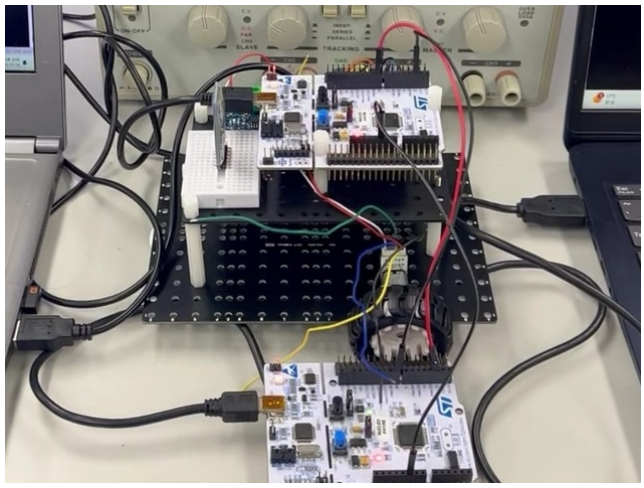
main.c

```
void USART1_IRQHandler(){
    if(is_USART1_RXNE()){
        PC_Data = USART1_read();

      if(PC_Data == 'L'){
            GPIO_write(GPIOA, LED_PIN, 0);
        }
        else if(PC_Data == 'H'){
            GPIO_write(GPIOA, LED_PIN, 1);
        }


      printf("MCU_1 received : %c \r\n",PC_Data); // TX to USART2(PC)

    }
}
```

USART1 can be used to communicate between MCUs. At this time, the code was written to turn off the other party's LED by entering L and turning the LED on by entering H.

Result

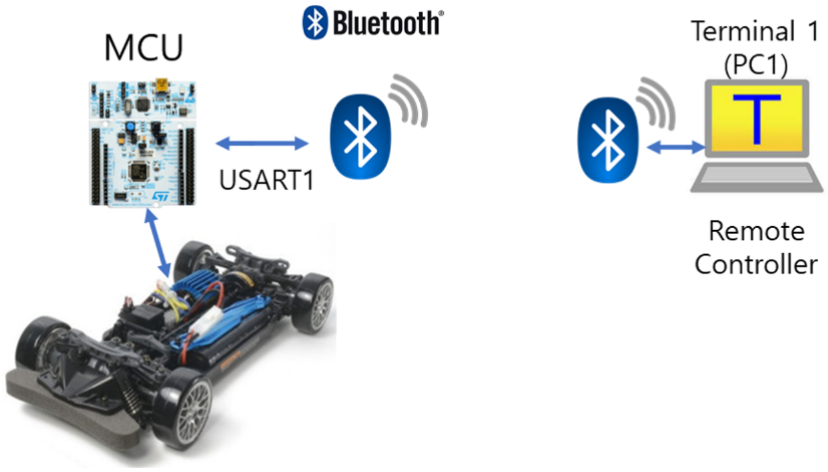
Entering L in Teratum turns off the other party's LED, while entering H turns it on. Similarly, when the other person inputs L or H, the LED on the MCU turns off or turns on.



Video Link:https://youtu.be/DysFVkn8NI4

Problem3 : Comtrol DC Motor via Bluetooth

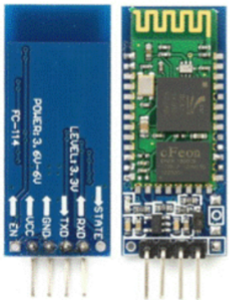
Bluetooth



image

Search for the bluetooth module specification sheet (HC-06) and study the pin configurations. The default PIN number is 1234.

Example of connecting to USART1



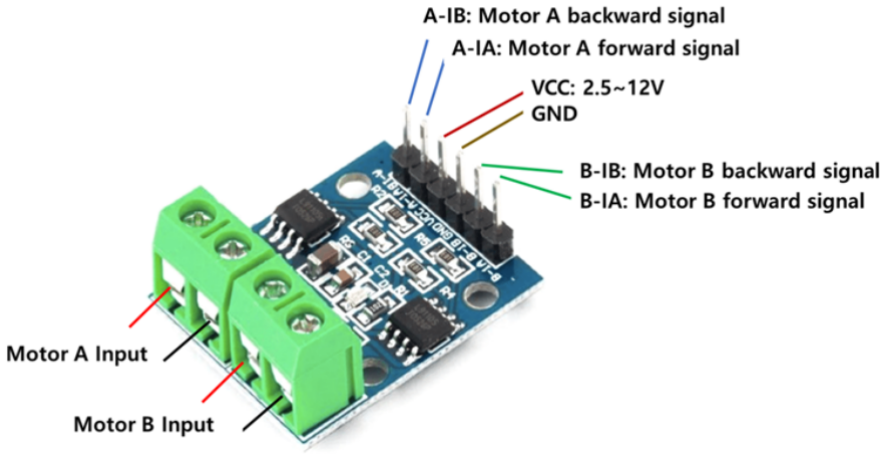
Bluetooth Module (HC-06)	STM32F411RE
RxD	PA_9(UART1_TX)
TxD	PA_10(UART1_RX)
GND	GND
VCC	5V

DC Motor Driver

Connect DC motor driver(L9110s) module pins to MCU as shown below.

DO NOT use MCU's VCC to motor driver. You should use external voltage source.

- A- IA: PWM pin (0~100% duty) for Motor A
- A- IB: Direction Pin (Digital Out H or L) for Motor B



Procedure

1. Create a new project under the directory `\repos\EC\LAB\LAB_USART_Bluetooth

- The project name is "**LAB_USART_Bluetooth".**
- Create a new source files named as "**LAB_USART_Bluetooth.c"**

2. Include your updated library in `\repos\EC\lib\` to your project.

- **ecGPIO.h, ecGPIO.c**
- **ecRCC.h, ecRCC.c**
- **ecUART.h, ecUART.c**
- **ecTIM.h, ecTIM.c**

3. Connect the MCU to PC via Bluetooth. Use USART 1

- connect RX/TX pins as
  - MCU TXD - BLUE RXD

- MCU RXD - BLUE TXD

4. Check the Bluetooth connection by turning MCU's LED(LD2) On/OFF by sending text of "**L0**" or "**L1**" from PC.

5. Run 2 DC motors(Left-wheel, Right-wheel) to steer.

- Turn Left: MotorA / MotorB = (50 / 80%) duty
- Turn Right: MotorA / MotorB = (80 / 50%) duty
- Go straight: MotorA / MotorB = (80 / 80 %) duty
- STOP: MotorA / MotorB = (0 / 0 %) duty

Configuration

Type	Port - Pin	Configuration
System Clock		PLL 84MHz
USART1 : MCU - Bluetooth	TXD: PA9 RXD: PA10	No Parity, 8-bit Data, 1-bit Stop bit, 9600 baud-rate
Digital Out: LD2	PA5	
PWM (Motor A)	TIM2-Ch1	PWM period (2kHz~10kHz)
PWM (Motor B)	TIM2-Ch2	

Code

```
int main(void){
  setup();
  printf("MCU Initialized\r\n");
  while(1){

    PWM_duty(PWM_PIN_1,Duty_1);
    PWM_duty(PWM_PIN_2,Duty_2);


    USART2_write(PC_string, 7);
    delay_ms(2000);
  }
}
```

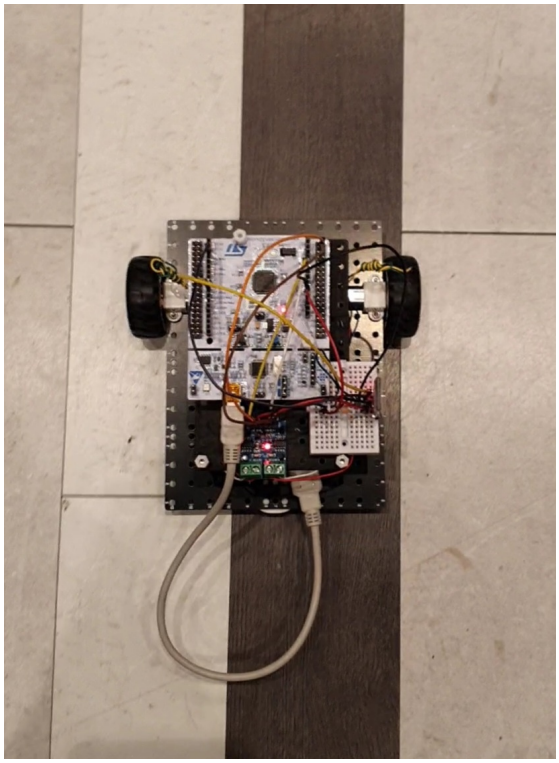
Prepared for motor control by using the PWM_duty function of both motors in the main function.

```
void USART1_IRQHandler(){
  if(is_USART1_RXNE()){
    PC_Data = USART1_read();
    USART_write(USART1, &PC_Data,1);

    if(PC_Data == 'W'){        //forward
      Duty_1=0.8;
      Duty_2=0.8;

  }
  else if(PC_Data == 'S'){  //stop

      Duty_1=0;
      Duty_2=0;
  }
  else if(PC_Data == 'A'){  //left
      Duty_1=0.8;
      Duty_2=0.5;
  }
  else if(PC_Data =='D') {  //right
      Duty_1=0.5;
      Duty_2=0.8;
  }

    printf("RX: %c \r\n",PC_Data);
  }
}
```

When W, S, A, and D are entered into Teratum, the code was written so that the RC car moves forward, stops, turns left, and turns right.

Result

The RC car was controlled by inputting W, S, A, D into Teartum.



Video Link: https://youtu.be/9s9FBUpjwYs

Reference

https://github.com/ykkimhgu

https://ykkim.gitbook.io/ec/ec-course/lab/lab-usart-led-bluetooth

Appendix

LED.c

```
#include "stm32f4xx.h"
#include "ecGPIO.h"
#include "ecRCC.h"
#include "ecUART.h"
#include "ecSysTick.h"


static volatile uint8_t PC_Data = 0;
static volatile uint8_t BT_Data = 0;
void LED_toggle();

void setup(void){
    RCC_PLL_init();

    // USB serial init
    UART2_init();
    UART2_baud(BAUD_38400);

    // BT serial init
    UART1_init();
    UART1_baud(BAUD_38400);

      //LED setup
      GPIO_init(GPIOA, LED_PIN, OUTPUT);     // calls RCC_GPIOA_enable()
}

void main(){
    setup();

    while(1){
    }
}

void USART2_IRQHandler(){                // USART2 RX Interrupt : Recommended
    if(is_USART2_RXNE()){
        PC_Data = USART2_read();       // RX from UART2 (PC)
        USART2_write(&PC_Data,1);       // TX to USART1     (BT)
            USART1_write(&PC_Data,1);
        printf("MCU_1 sent : %c \r\n",PC_Data); // TX to USART2(PC)
    }
}

void USART1_IRQHandler(){
    if(is_USART1_RXNE()){
         PC_Data = USART1_read();

      if(PC_Data == 'L'){
            GPIO_write(GPIOA, LED_PIN, 0);
         }
         else if(PC_Data == 'H'){
            GPIO_write(GPIOA, LED_PIN, 1);
         }

      printf("MCU_1 received : %c \r\n",PC_Data); // TX to USART2(PC)

    }
}
```

Bluetooth.c

```c
#define PWM_PIN_1 PA_0
#define PWM_PIN_2 PA_1
#define L_DIR 2
#define R_DIR 3

static volatile uint8_t PC_Data = 0;
static volatile uint8_t BT_Data = 0;
static volatile float Duty_1 = 0;
static volatile float Duty_2 = 0;


uint8_t PC_string[]="Loop:\r\n";

void setup(void){
  RCC_PLL_init();
  SysTick_init();

  // USART2: USB serial init
  UART2_init();
  UART2_baud(BAUD_38400);

  // USART1: BT serial init
  UART1_init();
  UART1_baud(BAUD_9600);

  //PWM init
  PWM_init(PWM_PIN_1);
  PWM_init(PWM_PIN_2);
  PWM_period(PWM_PIN_1,1);
  PWM_period(PWM_PIN_2,1);

  GPIO_init(GPIOC,L_DIR,OUTPUT);
  GPIO_write(GPIOC,L_DIR,LOW);
  GPIO_init(GPIOC,R_DIR,OUTPUT);
  GPIO_write(GPIOC,R_DIR,LOW);

}
```

```c
int main(void){
  setup();
  printf("MCU Initialized\r\n");
  while(1){

    PWM_duty(PWM_PIN_1,Duty_1);
    PWM_duty(PWM_PIN_2,Duty_2);


    USART2_write(PC_string, 7);
    delay_ms(2000);
  }
}
```

```
void USART1_IRQHandler(){
  if(is_USART1_RXNE()){
    PC_Data = USART1_read();
    USART_write(USART1, &PC_Data,1);

    if(PC_Data == 'W'){        //forward
      Duty_1=0.8;
      Duty_2=0.8;

  }
  else if(PC_Data == 'S'){  //stop

      Duty_1=0;
      Duty_2=0;
  }
  else if(PC_Data == 'A'){  //left
      Duty_1=0.8;
      Duty_2=0.5;
  }
  else if(PC_Data =='D') {  //right
      Duty_1=0.5;
      Duty_2=0.8;
  }

    printf("RX: %c \r\n",PC_Data);
  }
}
```