



## Lab8 UART and ADC

### 實驗八 UART 及 ADC

#### 1. Lab objectives 實驗目的

- Understand the use of UART.
- Understand the use of ADC.
- 瞭解 UART 的使用
- 瞭解 ADC 的使用

#### 2. Lab principle 實驗原理

- Please refer to the lecture Slides.
- 請參考上課講義。

baud rate: 資訊源上傳輸出 bits 的速度  
(包含 start, data, parity, stop bit...)  
bit rate: 資訊源帶來真正資訊的量的速度  
(只包含 data 的部分)

#### 3. Steps 實驗步驟

##### 3.1. Hello World!

###### Requirement:

When the blue button on the board is pressed (PC13), use the UART to transfer the "Hello World!" string to the computer. And it can be displayed on the serial monitor of the computer side (e.g. putty, screen, piety, MobaXterm ...).

- init needs to use GPIO.
- Understand the registers of UART and how to use them.
- Implement this function using the transmit of UART (TX)

在按下板子上藍色按鈕時 (PC13)，請利用 UART 將 "Hello World!" 字串傳送到電腦。並且可以在電腦端的 serial monitor (e.g. putty, screen, piety, MobaXterm ...) 顯示出來。

- init 需要使用到的 GPIO。
- 瞭解 UART 的暫存器以及使用方式
- 利用 UART 的傳出 (TX) 來實作此功能

```
main.c
```

```
int UART_Transmit(uint8_t *arr, uint32_t size) {  
    //TODO: Send str to UART and return how many bytes are successfully  
    transmitted.  
}  
  
void init_UART() {  
    // Initialize UART registers  
}
```



### 3.2. Photoresistor and ADC 光敏電阻及ADC

#### Requirement:

Use the ADC (Analog-to-Digital Converter) provided on the board to read the value of the photoresistor in 12-bit resolution using Interrupt, and use the UART output value each time the button (PC13) is pressed.

- Turn on the ADC and initialize its settings
- Each time the button is pressed, the UART is used to transmit the photoresistor value.

請利用板子上提供的 ADC (Analog-to-Digital Converter) 利用 Interrupt 將光敏電阻的值以 12-bit 的解析度讀出，並且每按一次按鈕 (PC13) 時利用 UART 輸出數值。

- 開啟 ADC 並且初始化其設定
- 每次按下按鈕利用 UART 傳輸光敏電阻值出去

main.c

```
void configureADC() {  
    // TODO setting ADC register  
}  
void startADC() {  
    // TODO enable ADC  
}
```

### 3.3. Simple Shell

#### Requirement:

Implement Simple Shell on the board using UART and have three instructions 在板子上利用 UART 實作 Simple Shell，並且擁有三個指令

1. showid
  - a. show your student ID
2. light
  - a. Update and display the value of the photoresistor every 0.5 seconds (lab10.2), press 'q' to return to the original shell mode
3. led < on | off >
  - a. "led on" will turn the PA5 LED on, "led off" will turn off

After the USB to UART is connected to the computer, any characters entered by the computer must be received through the board's UART RX and transmitted back to the computer using TX to achieve echo. The computer runs a serial monitor (e.g. putty, piety, MobaXterm ...) to use the shell.



在電腦端連接 USB to UART 後，任何電腦輸入的字元必須透過板子的 UART RX 收到並且在利用 TX 傳回給電腦達到 echo 的效果。電腦端跑 serial monitor (e.g. putty, pietty, MobaXterm ...)去使用該 shell.

**Note:**

- Do ***not*** implement the shell on the computer, it must be on the STM32L476RG
  - Can use <string.h> and <stdlib.h>
  - Implement UART\_Receive function
  - '>' as the prompt character
  - For any instruction other than these three, please display "Unknown Command"
  - The length of the TA test command will be between 1 and 100.
  - If you press Enter without entering any command, the next prompt character will be displayed.
  - Must be implemented backspace
- 
- 請勿在電腦端上實作 Shell，必須在 STM32L476RG 上
  - 可以使用 <string.h> 及 <stdlib.h>
  - 實作 UART\_Receive
  - 統一以 '>' 為提示字元
  - 任意這三個以外的指令，請皆顯示「Unknown Command」
  - 助教測試指令的長度會介在 1 到 100 之間
  - 未輸入任何指令直接按 Enter 的話會顯示下一個提示字元
  - 需要實作 backspace

```
main.c
```

```
int UART_Receive(char *c) {  
    // TODO receive one character from UART  
}
```