實驗九 **UART**、**LCD**、**ADC**

1. 實驗目的

* 瞭解UART的使用
* 瞭解LCD的使用
* 瞭解ADC的使用

1. 實驗原理

請參考上課講義。

1. 實驗步驟
   1. **S****end to LCD (50%)**

電腦端透過UART傳送字串給stm32，stm32接收到字串後顯示在LCD上，並且透過UART回傳該字串加上” showed”

* 舉例：在電腦輸入”Hey”，LCD需顯示”Hey”，並且電腦顯示回傳字串為”Hey showed” (以上舉例皆不包括雙引號)
* 測資只會有大小寫英文，長度不會超過8個英文字

* 1. 光敏電阻讀值 **(50%)**

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

* 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  }  void startADC()  {  // TODO  } |

1. 實驗結果

4.1.Send to LCD

此實驗需要透過UART讓電腦與板子互相溝通。其中Tx(A9)為輸出(板子<-電腦)端，Rx(A10)為接收(板子->電腦)端。

注意事項：

1.需將pin A9,A10設為Alternative Function Mode

2.Baudrate為9600(隨便設的)

3.Tx接Rx，Rx接Tx

4.2. 光敏電阻讀值

4-2比起4-1，還需多ADC的操作（Code如下圖）。

ADC(Analog to Digital Convertor)是板子內建的簡易轉換器，其會將光敏電阻(Analog)的輸出當作自己的輸入，並輸出Digital的訊號給CPU。ADC每隔一段時間就會將輸入端的訊號轉為數位，此部分是透過定時Interrupt來實作。當使用者按下UserButton時，會對板子造成外部中斷(EXTI)，此時板子便將ADC轉換好的訊號透過UART傳給電腦。

|  |
| --- |
| **void init\_GPIO(){**  **RCC->AHB2ENR |= 7;//Enable Port A B C**  **GPIOA->MODER = 0xABEBFFFF;**  **GPIOA->AFR[1] = 0x770; // AF7 for pin**  **//Set PC13 INPUT mode & PC0 analog mode**  **GPIOC->MODER &= 0xF3FFFFFF;**  **GPIOC->MODER |= 0x3;**  **GPIOC->ASCR = 0x1;**  **}** |

|  |
| --- |
| **void USART1\_Init(void) {**  **/\* Enable clock for USART??? \*/**  **RCC->APB2ENR |= RCC\_APB2ENR\_USART1EN;**  **// CR1**  **USART1->CR1 &= 0x0;**  **USART1->CR1 |= (USART\_CR1\_TE | USART\_CR1\_RE);**  **//MODIFY\_REG(USART1->CR1, USART\_CR1\_M | USART\_CR1\_PS | USART\_CR1\_PCE | USART\_CR1\_TE | USART\_CR1\_RE |USART\_CR1\_OVER8, USART\_CR1\_TE | USART\_CR1\_RE);**  **// CR2**  **USART1->CR2 &= 0x0;**  **USART1->CR2 |= (0x00 << 12);**  **//MODIFY\_REG(USART1->CR2, USART\_CR2\_STOP, 0x0); // 1-bit stop**  **// CR3**  **USART1->CR3 &= 0x0;**  **USART1->CR3 |= (0x0<<11);**  **//MODIFY\_REG(USART1->CR3, (USART\_CR3\_RTSE | USART\_CR3\_CTSE | USART\_CR3\_ONEBIT), 0x0); // none hwflowctl**  **USART1->BRR &= 0x0;**  **USART1->BRR |= 417L;//9600L;//4000000L/417L;**  **//MODIFY\_REG(USART1->BRR, 0xFF, 4000000L/???? L);**  **/\* In asynchronous mode, the following bits must be kept cleared:**  **- LINEN and CLKEN bits in the USART\_CR2 register,**  **- SCEN, HDSEL and IREN bits in the USART\_CR3 register.\*/**  **USART1->CR2 &= ~(USART\_CR2\_LINEN | USART\_CR2\_CLKEN);**  **USART1->CR3 &= ~(USART\_CR3\_SCEN | USART\_CR3\_HDSEL | USART\_CR3\_IREN);**  **// Enable UART**  **USART1->CR1 |= (USART\_CR1\_UE);**  **}** |

注意事項：

1.要將接收光敏電阻值的pin (PC0)設定為analog mode，並設為ADC輸入端

2.光敏電阻要串接電阻，避免燒毀

3.ADC1，Channel1

|  |
| --- |
| **void configureADC(){**  **// TODO**  **RCC->AHB2ENR |= RCC\_AHB2ENR\_ADCEN;**  **ADC1->CR &= ~ADC\_CR\_DEEPPWD;// Clear Deep Sleep**  **//CLEAR\_BIT(ADC1->CR, ADC\_CR\_DEEPPWD);**  **ADC1->CR |= ADC\_CR\_ADVREGEN;//SET\_BIT(ADC1->CR, ADC\_CR\_ADVREGEN);**  **// Turn on Voltage Regulator**  **delay\_us(200);**  **// Prescaler**  **ADC123\_COMMON->CCR &= ~(ADC\_CCR\_PRESC|ADC\_CCR\_CKMODE);**  **ADC123\_COMMON->CCR |= ADC\_CCR\_CKMODE\_0;**  **//MODIFY\_REG(ADC123\_COMMON->CCR, ADC\_CCR\_PRESC|ADC\_CCR\_CKMODE, ADC\_CCR\_CKMODE\_0);**  **//ADC1->CFGR &= ~(ADC\_CFGR\_FIELDS\_1);**  **ADC1->CFGR |= ADC\_CFGR\_CONT;**  **//MODIFY\_REG(ADC1->CFGR, ADC\_CFGR\_FIELDS\_1, ADC\_CFGR\_CONT);**  **ADC1->CFGR2 &= ~ ADC\_CFGR2\_ROVSE;**  **//CLEAR\_BIT(ADC1->CFGR2, ADC\_CFGR2\_ROVSE);**  **ADC1->SQR1 &= ~ADC\_SQR1\_L;**  **//CLEAR\_BIT(ADC1->SQR1, ADC\_SQR1\_L);**  **ADC1->SQR1 &= ~((0xF)|(0x1F<<24)|(0x1F<<18)|(0x1F<<12)|(0x1F<<6));**  **ADC1->SQR1 |= (0x1<<6);// Channel 1, Rank 1**  **//MODIFY\_REG(ADC1->SQR1, (0xF)|(0x1F<<24)|(0x1F<<18)|(0x1F<<12)|(0x1F<<6), (0x1<<6)); // Channel 1, Rank 1**  **ADC1->SMPR1 &= ~ (0x3FFFFFFFFF);**  **ADC1->SMPR1 |= (0x6<<3);// Channel 1, Sampling Time: 247.5 ADC cycles**  **//MODIFY\_REG(ADC1->SMPR1, (0x3FFFFFFFFF), (0x6<<3)); // Channel 1, Sampling Time: 247.5 ADC cyc**  **ADC1->IER = (1<<ADC\_IER\_ADRDYIE);// ADC ready interrupt enable**  **NVIC\_SetPriority(ADC1\_2\_IRQn, 1);**  **NVIC\_EnableIRQ(ADC1\_2\_IRQn);**  **}** |

|  |
| --- |
| **void startADC() {**  **while (!(ADC1->ISR & ADC\_ISR\_ADRDY)) ADC1->CR |= ADC\_CR\_ADEN; // TURN ON**  **delay\_us(5000);**  **ADC1->ISR = ADC\_ISR\_EOC | ADC\_ISR\_EOS | ADC\_ISR\_OVR; // Clear flags**  **//interrupt and status register**  **ADC1->CR |= ADC\_CR\_ADSTART;// START CONV**  **//ADC control register |= ADC start of regular conversion**  **//SET\_BIT(ADC1->CR, ADC\_CR\_ADSTART);**  **}** |

5.心得

Code寫好了，線也接好，看I/O reg確實有值傳過去，不過不知道為什麼terminal上的視窗都沒有結果。後來到實驗室問同學才知道，原來是Tx Rx要交換接，助教的pdf都沒寫，  
我也沒來聽lab，不知道QQ。幸好是最後一次lab了，真高興～