

Câu chuyện bắt đầu từ một cậu bé,
và một ý tưởng
có thể
làm thay đổi thế giới...

PAY IT FORWARD

Đó là khi bạn giúp đỡ 3 người bạn không quen biết,
dù là bằng thời gian,
hay công sức,
hay kinh nghiệm,
hay kiến thức,
hay tiền bạc, ...
của mình.



Mà không chờ đợi một sự báo ân nào.

Chỉ cần mỗi người trong 3 người đó,
lại đem những gì mình có, mà người khác cần,
tiếp tục giúp đỡ thêm 3 người nữa.

Chính những người-giúp-đỡ, và người-được-giúp-đỡ,
sẽ là những người góp phần thay đổi thế giới...

Một thế giới sẽ chia kiến thức - và yêu thương ...

PAY IT FORWARD ...

Chúng tôi không sáng tạo ra câu nói này.

Pay it forward...

Hãy tri ân người giúp mình bằng cách giúp đỡ người khác
Cho đi không phải để nhận lại.

PAY IT FORWARD



Universal Asynchronous Receiver/Transmitter (UART)

28/10/2014

1. Tổng quan UART

2. Sơ đồ khối

3. Mô tả chức năng

4. Cài đặt UART

5. Chương trình mẫu
và thảo luận



Tổng quan UART

★ Separate 16x8 bit transmit and receive FIFOs

- ✓ Thanh ghi dữ liệu UARTDR
- ✓ Set bit FEN trong thanh ghi UARTLCRH
- ✓ Thanh ghi cờ UARTFR và thanh ghi trạng thái nhận UARTRSR



Tổng quan UART

- ★ Separate 16x8 bit transmit and receive FIFOs
- ★ Programmable baud rate generator

- ✓ Số 22 bit: 16 bit phần nguyên và 6 bit phần thập phân.
- ✓ $BRD = BRDI + BRDF$
 $= \text{UARTSysClk} / (\text{ClkDiv} * \text{Baud Rate})$
 $\text{ClkDiv} = 16$, bit HSE trong UARTCTL xóa.
 $\text{ClkDiv} = 8$, bit HSE trong UARTCTL set



Tổng quan UART

- ★ Separate 16x8 bit transmit and receive FIFOs
- ★ Programmable baud rate generator
- ★ Line break generation and detection
- ★ Modem flow control on UART1 (RTS/CTS)

- ✓ Hardware: CTS và RTS
- ✓ Software: UARTTIM, UARTRIS, UARTMIS, UARTICR



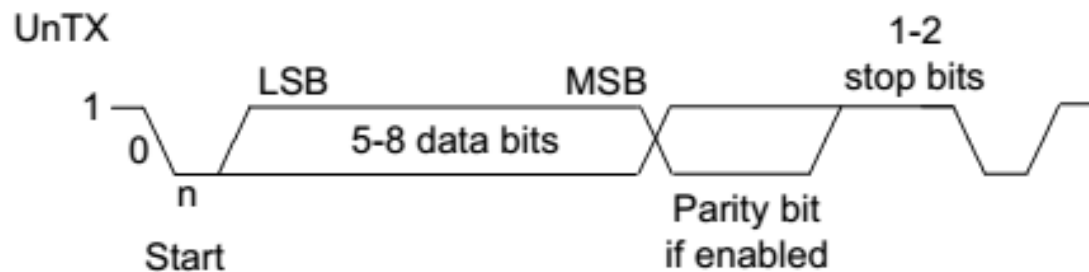
Tổng quan UART

- ★ Separate 16x8 bit transmit and receive FIFOs
- ★ Programmable baud rate generator
- ★ Line break generation and detection
- ★ Modem flow control on UART1 (RTS/CTS)
- ★ μ DMA support
- ★ EIA-495 9-bit operation



Tổng quan UART

- ★ Programmable serial interface
 - 5, 6, 7, or 8 data bits
 - Even, odd, stick, or no parity bits
 - 1 or 2 stop bits

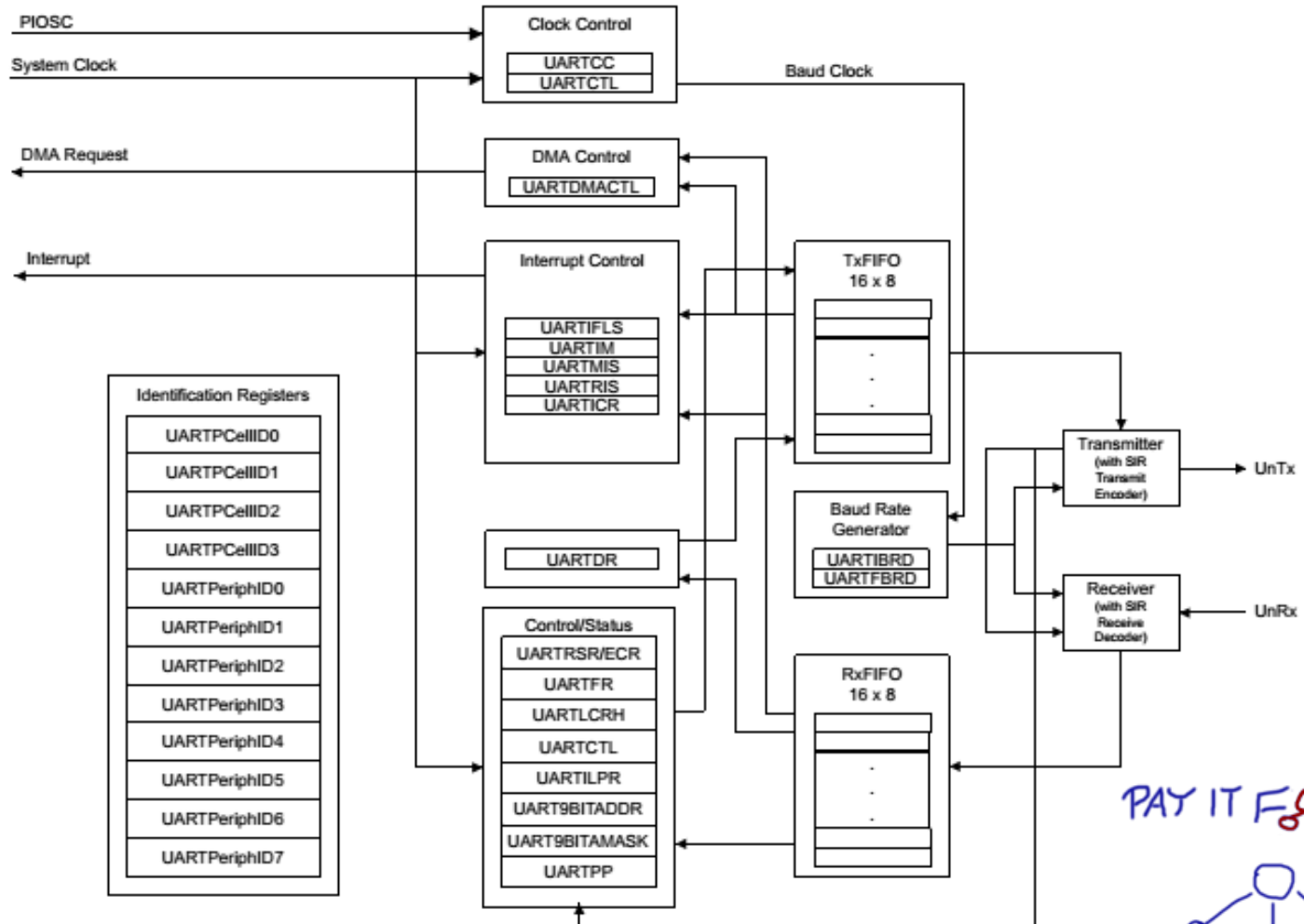


- ★ IrDA serial IR (SIR) encoder/decoder
- ★ ISA 7816 smartcard support
- ★ LIN support: master or slave

PAY IT FORWARD



Sơ đồ khối

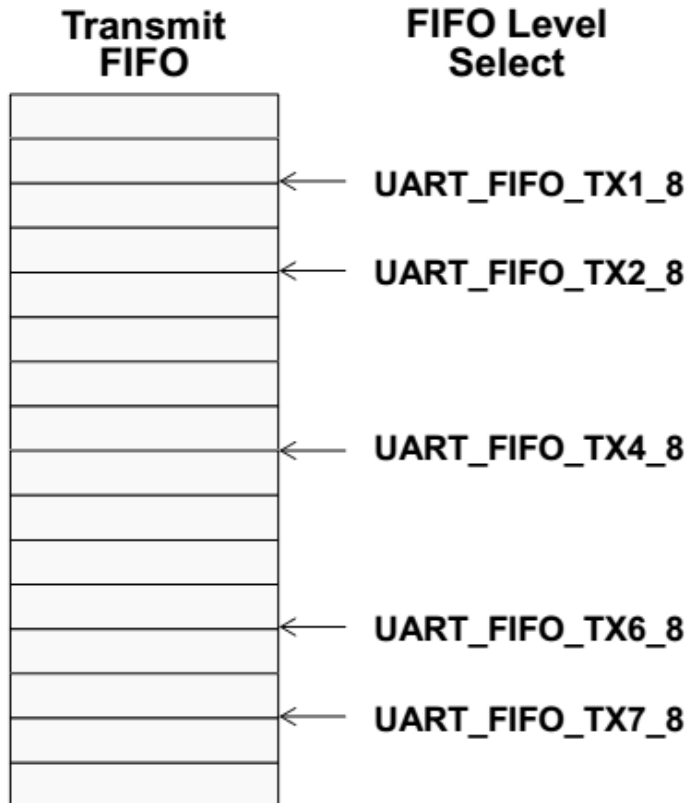


Ngắt UART

- ★ Single interrupt per module, cleared automatically
- interrupt conditions:
- Overrun error
 - Break error
 - Parity error
 - Framing error
 - Receive timeout
 - Transmit – generated when no data present
 - Receive – generated when character is received



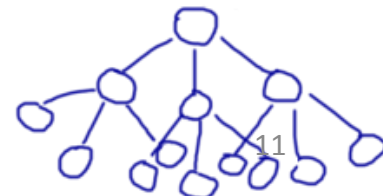
Chế độ FIFO



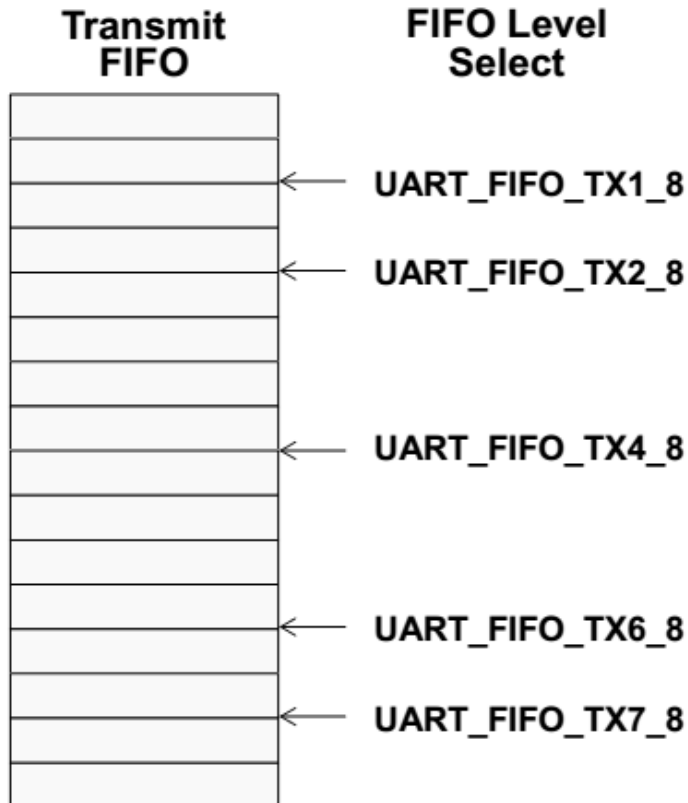
- ★ Tất cả các FIFOs được truy cập qua thanh ghi dữ liệu UART (UARTDR)
- ★ Sau khi reset, Các FIFOs sẽ được enabled, có thể disable bằng cách reset bit FEN trong thanh ghi UARTLCRH.

UARTFIFODisable (UART0_BASE) ;

PAY IT FORWARD



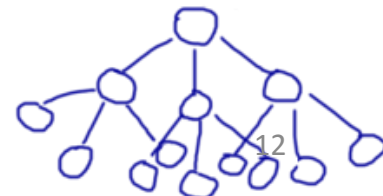
Chế độ FIFO



- ★ Trigger points cho ngắt FIFO có thể được set 1/8, 1/4, 1/2, 3/4, 7/8 full

```
UARTFIFOLevelSet (UART0_BASE,  
UART_FIFO_TX4_8,  
UART_FIFO_RX4_8);
```

PAY IT FORWARD



Cài đặt UART

★ Cấu hình UART

▪ Khởi động UART

```
SysCtlPeripheralEnable (SYSCTL_PERIPH_UART0);
```

```
SysCtlPeripheralEnable (SYSCTL_PERIPH_GPIOA);
```

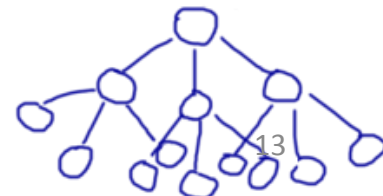
▪ Set chân Rx/Tx thành chân sử dụng UART

```
GPIOPinConfigure (GPIO_PA0_U0RX);
```

```
GPIOPinConfigure (GPIO_PA1_U0TX);
```

```
GPIOPinTypeUART (GPIO_PORTA_BASE, GPIO_PIN_0 |  
GPIO_PIN_1);
```

PAY IT FORWARD



Cài đặt UART

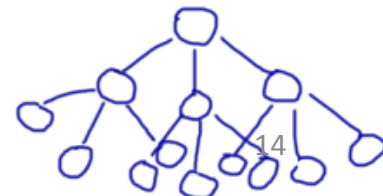
★ Cấu hình UART

- Thiết lập baud rate, kiểu dữ liệu cho UART

```
UARTConfigSetExpClk(UART0_BASE, SysCtlClockGet(),  
115200, (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE  
| UART_CONFIG_PAR_NONE));
```

- Thiết lập các đặc tính khác của UART (e.g. interrupts, FIFO)

PAY IT FORWARD



Cài đặt UART

★ Gửi và nhận kí tự

- Blocking/non-blocking function in driverlib:

```
UARTCharPut (UART0_BASE, 'a' );  
Newchar = UARTCharGet (UART0_BASE);  
UARTCharPutNonBlocking (UART0_BASE, 'a' );  
Newchar = UARTCharGetNonBlocking (UART0_BASE);
```



Chương trình mẫu – Ngắt UART

- ★ Thêm các file header .h cho ngắt UART

```
#include "inc/hw_ints.h"  
#include "driverlib/interrupt.h"
```

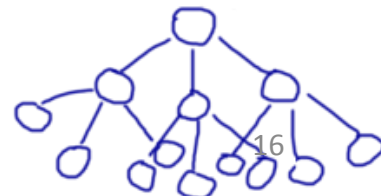
- ★ Cho phép ngắt toàn cục, ngắt UART

```
IntMasterEnable(); //enable processor interrupts  
IntEnable(INT_UART0); //enable the UART interrupt  
UARTIntEnable(UART0_BASE, UART_INT_RX |  
UART_INT_RT); //only enable RX and TX interrupts
```

- ★ Cài đặt chân cho ngoại vi

```
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);  
//enable GPIO port for LED  
GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_2);  
//enable pin for LED PF2
```

PAY IT FORWARD



Chương trình mẫu – Ngắt UART

```
#include <stdint.h>
#include <stdbool.h>
#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/gpio.h"
#include "driverlib/interrupt.h"
#include "driverlib/pin_map.h"
#include "driverlib/sysctl.h"
#include "driverlib/uart.h"
```



Chương trình mẫu – Ngắt UART

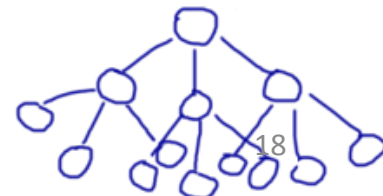
```
void UARTIntHandler(void)
{
    uint32_t ui32Status;

    ui32Status = UARTIntStatus(UART0_BASE, true); //get interrupt status

    UARTIntClear(UART0_BASE, ui32Status); //clear the asserted
    interrupts

    while(UARTCharsAvail(UART0_BASE)) //loop while there are chars
    {
        UARTCharPutNonBlocking(UART0_BASE,
        UARTCharGetNonBlocking(UART0_BASE));
        //echo character
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, GPIO_PIN_2); //blink LED
        SysCtlDelay(SysCtlClockGet() / (1000 * 3)); //delay ~1 msec
        GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0); //turn off LED
    }
}
```

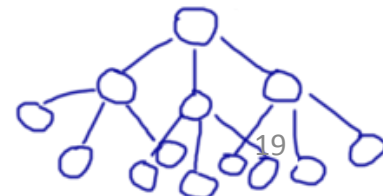
PAY IT FORWARD



Chương trình mẫu – Ngắt UART

```
int main(void) {  
  
    SysCtlClockSet(SYSCTL_SYSDIV_4 | SYSCTL_USE_PLL | SYSCTL_OSC_MAIN |  
SYSCTL_XTAL_16MHZ);  
  
    SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);  
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);  
  
    GPIOPinConfigure(GPIO_PA0_U0RX);  
    GPIOPinConfigure(GPIO_PA1_U0TX);  
    GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);  
  
    SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF); //enable GPIO port for  
LED  
    GPIOPinTypeGPIOOutput(GPIO_PORTF_BASE, GPIO_PIN_2); //enable pin for  
LED PF2  
  
    UARTConfigSetExpClk(UART0_BASE, SysCtlClockGet(), 115200,  
        (UART_CONFIG_WLEN_8 | UART_CONFIG_STOP_ONE |  
UART_CONFIG_PAR_NONE));  
}
```

PAY IT FORWARD



```

IntMasterEnable(); //enable processor interrupts
IntEnable(INT_UART0); //enable the UART interrupt
UARTIntEnable(UART0_BASE, UART_INT_RX | UART_INT_RT); //only enable RX and
TX interrupts
UARTCharPut(UART0_BASE, 'E');
UARTCharPut(UART0_BASE, 'n');
UARTCharPut(UART0_BASE, 't');
UARTCharPut(UART0_BASE, 'e');
UARTCharPut(UART0_BASE, 'r');
UARTCharPut(UART0_BASE, ' ');
UARTCharPut(UART0_BASE, 'T');
UARTCharPut(UART0_BASE, 'e');
UARTCharPut(UART0_BASE, 'x');
UARTCharPut(UART0_BASE, 't');
UARTCharPut(UART0_BASE, ':');
UARTCharPut(UART0_BASE, ' ');
while (1) //let interrupt handler do the UART echo function
{
    // if (UARTCharsAvail(UART0_BASE)) UARTCharPut(UART0_BASE,
UARTCharGet(UART0_BASE));
}
}

```

PAY IT FORWARD

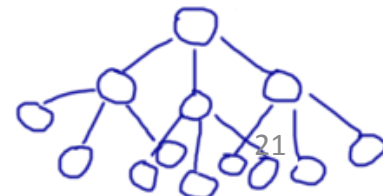
Chương trình mẫu – Ngắt UART

- ★ Bước cuối cùng, chúng ta cần chỉnh sửa file **tm4c123gh6pm_startup_ccs.c** trước khi chạy chương trình
 - Mở file **tm4c123gh6pm_startup_ccs.c**
 - Ta sử dụng bao nhiêu tác vụ ngắt thì phải khai báo sau dòng **extern void _c_int00(void);**
 - Trong trường hợp này ta sẽ thêm dòng sau:

```
extern void _c_int00(void);
```

```
extern void UARTIntHandler(void);
```

//Thêm dòng này



Chương trình mẫu – Ngắt UART

★ Thay dòng:

IntDefaultHandler, // UART0 Rx and Tx

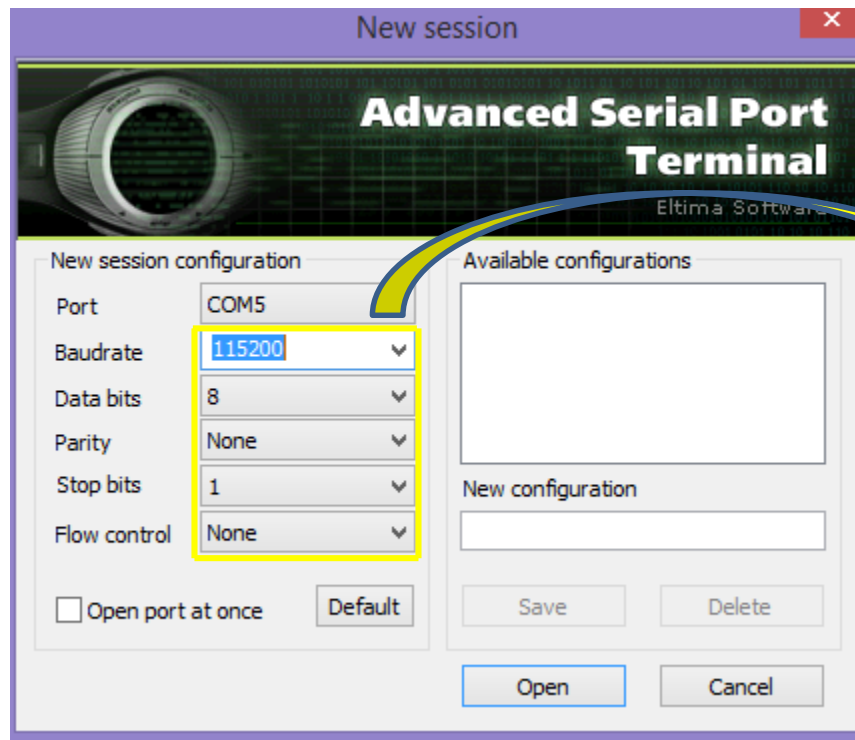
Thành:

UARTIntHandler, // UART0 Rx and Tx



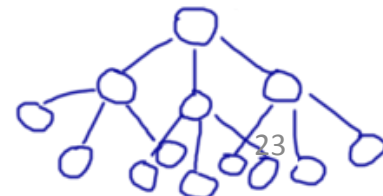
Chương trình mẫu – Ngắt UART

- ★ Sử dụng phần mềm **Advance Serial Port Terminal** để xem dữ liệu được gửi ra cổng COM thông qua UART



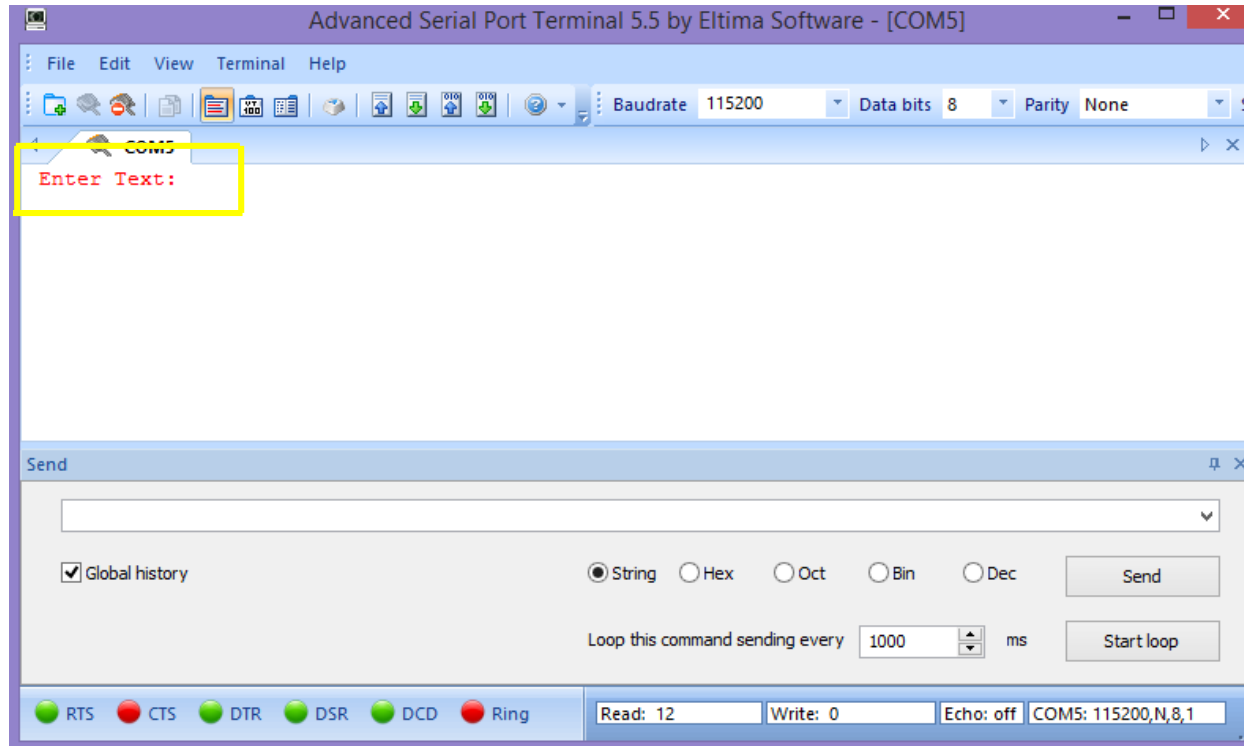
```
UARTConfigSetExpClk(UART0_
BASE, SysCtlClockGet(),
115200,
    (UART_CONFIG_WLEN_8 |
UART_CONFIG_STOP_ONE |
UART_CONFIG_PAR_NONE));
```

PAY IT FORWARD

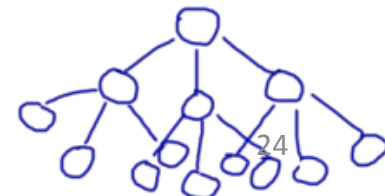


Chương trình mẫu – Ngắt UART

★ Kết quả hiển thị:



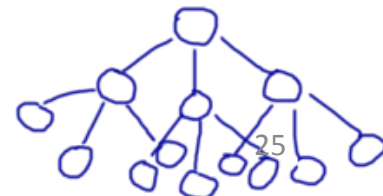
PAY IT FORWARD

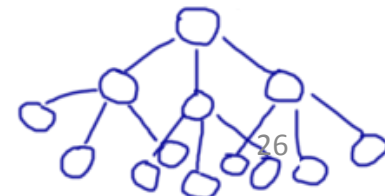


Chương trình mẫu – Ngắt UART

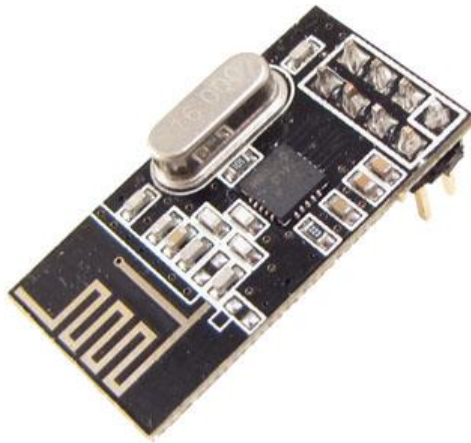
- ★ Code được dùng để gửi các kí tự dùng hàm `UARTCharPut()` đã được viết sẵn trong "`driverlib/uart.h`"
- ★ Khi nhập kí tự vào ô SEND, chúng ta sẽ thấy kí tự hiện thị trên cổng COM ảo, đồng thời đèn led rgb tại Port F cũng sáng màu lam rồi tắt, nghĩa là dữ liệu gửi về TIVA đã được nhận thành công và gửi trở lại,
 - Trong hàm `while(1)`, bỏ "//" trước
`if (UARTCharsAvail (UART0_BASE))`
`UARTCharPut (UART0_BASE, UARTCharGet (UART0_BASE)) ;`
 - Khi thực hiện dòng while này, chương trình sẽ không thể nhảy vào ngắt, kí tự vẫn được echo lên COM ảo nhưng đèn led rgb không sáng.

PAY IT FORWARD





Các module sử dụng UART



NRF 24L01



Module SIM900



ez430-RF2500

Thảo luận

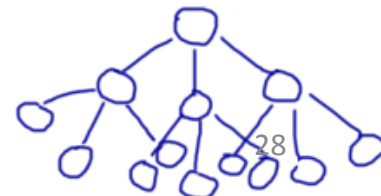
Làm thế nào để
gửi và nhận chuỗi
kí tự đây?

GO FOR IT !



GOOD LUCK !

PAY IT FORWARD



References

- [1] Peripheral Driver Library.pdf
- [2] TM4C123G_LaunchPad_Workshop_Workbook.pdf
- [3] tm4c123gh6pge.pdf



PAY IT FORWARD



payitforward.edu.vn