University of Dhaka

Department of Computer Science and Engineering

CSE-3116 : Microcontroller Lab

Lab Report 2 : Traffic Signal Management via USART

Submitted By:

Name : Asfin Jannat Shamsi

Roll No : 23

Name : Tanzeem Maliat

Roll No : 51

Submitted On :

April 9, 2023

Submitted To :

Dr. Mosaddek Hossain Kamal Tushar

Redwan Ahmed Rizvee

1 Introduction:

The objectives of the lab assignment are to understand and have hands-on training in microcontroller (STM32F446RE) UART (Universal Asynchronous Receiver Transmitter) and TIMER configuration and the operation of UART pins and ports as buffered input and output (using interrupt subroutine).

1.1 Objectives:

In this lab assignment, we have to design a central traffic monitoring system which will take configuration commands from the command line and configure traffic light intervention time along with extended time intervention in case of heavy traffic load.For this purpose, we have certain objectives.

- Configuration of UART/USART and GPIO pins and ports
- Configuration of UART/USART communication
- Configuration of TIMER to properly handle interrupt subroutine while communicating over UART/USART pins and ports

2 Assignment Problem Description:

In our country, traffic police usually control road traffic; however, the government has policies to alleviate our country into a digital and gradually smart Bangladesh. Thus it (police hand controlled) is against the policy towards smart Bangladesh. In affinity with the previous lab assignment, we plan to develop an automated system that periodically reports the traffic status with a configurable interval. Such as, the control center can define or change the interval to five seconds; let us assume it was three seconds previously, create delay and feed traffic to the monitor center according to the recent configuration.

3 Solution Approach:

For this assignment,we ignore the intermediate TrafficNet and wireless modem for the communication between control center and monitor center; instead,we directly connect

$$Tx_a \rightarrow Rx_b$$

,

$$Tx_b \rightarrow Rx_a$$

two UART ports of the microcontroller.

In addition,we are the current UART2 to display traffic information, including traffic signal light status, on the PC display (traffic monitoring system). We are also using UART4 as control center and UART5 as monitor centert to transmit and receive traffic data or configure the traffic-signal interval and transmission to the mechanism. Our developed system must transfer the configuration to UART5 through UART4, and receive monitor data on UART4 from UART5. The UART4 receives configuration commands from the microcontroller delivered through UART2 (currently available).

Furthermore, UART4 generated a monitoring report based on data from UART5 and sent them to the PC display system through UART2.

4 Technical Specifications:

4.1 Configuration Of GPIOx ports and pins:

For configuration of GPIOx:

```
void GPIO_Init(GPIO_TypeDef* GPIOx, GPIO_InitTypeDef *GPIO_Init)
{
        uint32_t i;
        uint32_t temp = 0x00U;

  /* Configure the port pins */
   for(i = 5; i < 9; i++)
   {
       /* Configure IO Direction mode (Input, Output, Alternate or Analog) */
       temp = GPIOx->MODER;
       temp &= ~(GPIO_MODER_MODER0 << (i * 2U));
       temp |= ((GPIO_Init->Mode & GPIO_MODE) << (i * 2U));
       GPIOx->MODER = temp;
   }
}
```

For writing the pin to SET or to RESET:

```
void GPIO_WritePin(GPIO_TypeDef *GPIOx, uint16_t GPIO_pin, GPIO_PinState PinState)
{

  if(PinState != GPIO_PIN_RESET)
  {
    GPIOx->BSRR |= GPIO_pin;
   }
   else
  {
    GPIOx->BSRR |= (uint32_t) GPIO_pin << 16U;
   }
}
```

4.2 Configuration Of USARTx/UARTx ports and pins:

We defined UART4 as control center which will take the configuration command lines from Hercules(PC to USART communication) and UART5 as monitor center where traffic light time intervention for red light of each

road is included via USART2 and UART4 and traffic light and status is stored Here, we are including the configuration of USART2 . The UART4 and UART5 will follow the same pattern : the only difference will be their respentive ports and register numbers.

```
void UART2_Config(void){

    //1. Enable UART clock and GPIO clock
    RCC->APB1ENR |= (1<<17); //enable UART 2
    RCC->AHB1ENR |= (1<<0); //enable GPIOA clock


    //2. Configure UART pin for Alternate function
    GPIOA->MODER |= (2<<4);
    //bits [5:4] -> 1:0 -->Alternate function for pin PA2
    GPIOA->MODER |= (2<<6);
    //bits [7:6] -> 1:0 -->Alternate function for PA3

    GPIOA->OSPEEDR |= (3<<4) | (3<<6);
    //bits [5:4] -> 1:1 -> high speed PA2; bits [7:6] -> 1:1 -> high speed PA3

    GPIOA->AFR[0] |= (7<<8);
    //Bytes (11:10:09:08) = 0:1:1:1 --> AF7 Alternate function for USART2
    //at pin PA2
    GPIOA->AFR[0] |= (7<<12);
    //Bytes (15:14:13:12) = 0:1:1:1 --> AF7 Alternate function for USART2
    //at pin PA3

    //3. Enable UART on USART_CR1 rgister
    USART2->CR1 = 0x00; //clear USART

    USART2->CR1 |= (1<<13);  // UE-bit enable USART

    //4. Program M bit in USART CR1 to define the word length
    USART2->CR1 &= ~(1U<<12); // set M bit  = 0 for 8-bit word length

    //5. Select the baud rate using the USART_BRR register.
    USART2->BRR |= (7<<0) | (24<<4); //115200

    //  6. Enable transmission TE and recieption bits in USART_CR1 register
```

```
        USART2->CR1 |= (1<<2); // enable RE for receiver
        USART2->CR1 |= (1<<3); //enable TE for transmitter

    USART2->CR1 |= USART_CR1_RXNEIE;
}
```

4.3 Configuration Of TIMx:

For properly handling interrupt subroutine, the configuration of TIM6 has been used.

```
    void TIM6Config(void){
         /* Enable Timer6 Clock*/
         RCC->APB1ENR |= (1<<4);

        /*Set the prescaler and the ARR*/
        TIM6->PSC = 9000-1;
        TIM6->ARR = 10000-1;
        TIM6->DIER |=TIM_DIER_UIE;

        NVIC_EnableIRQ(TIM6_DAC_IRQn);
        /*Enable the Counter of the Timer and wait for the Update Flag to set*/
        TIM6->CR1 |= TIM_CR1_CEN;
}
```

Then for the interrupt subroutine to be invoked each ***in*** time which is the configurable traffic monitoring time, ***TIM6_DAC_IRQHandler()*** function has been used.

```
void TIM6_DAC_IRQHandler(void)
{
          if(TIM6->SR & (TIM_SR_UIF))
                {
                                TIM6->SR &=~(TIM_SR_UIF);
                                totaltime++;
                                if((totaltime % in)==0)
                                        display();
                }
}
```

5 Logic and Circuit:

5.1 Logic:

For the logical thinking for the solution of the given assignment, we determined that there are four roads: north-south :: Road_1 and east-west :: Road_2. For north-south roads we have assigned transparent Red_Road_1 and for east-west roads we have assigned transparent Red_Road_2.

Here,traffic signal time will be given as a command line from Hercules which is R1 and R2. If there is an imbalance of traffic load i.e., heavy-light or light-heavy, an extension time T1 and T12 is also given in the ***configure traffic light*** As the traffic signals are transparent, only two sets of traffic signals have been used. Here we have also illustrated the real life based traffic density on the roads. The loads of the traffic density will have impact on the traffic signals.

For example,if Red_Road_1 is turned red, the signal will look for the load of the traffic density on the Road_2. If the load is low or medium, the Traffic_Road_1 will remain red for R1. On the other hand, if the load is high, the Red_Road_1 will be set for R1 plus T1 . Then it will turn off so that the high load will decrease by allowing the traffic to move forward or left.

In the previously stated scenario, we have only stated the status of the Red_Road_1. As for the Red_Road_2, it will remain off for R2 at first if the load of Road_1 is high. On the other hand, it will show the red signal for R2 plus T2 if the load of the traffic density is low or medium on Road_1.

5.2 Circuit:

We have allocated pins from STM32F446RET6 to represent the red lights and traffic of Road_1 and Road_2.

Red_Road_1	Traffic_Road_1	Red_Road_2	Traffic_Road1_2
PA5	PA6	PA7	PA8

We have used the reference manual to properly configure the ports and pins of the UARTx/USARTx in the circuit board for the communication and traffic management.For the circuit set up, we have used the conventional direction which is relevant to our logical approach. We have assigned Red_Road_1 (PA5) for the north-south roads as the traffic signal and Red_Road_2 (PA7) for the east-west roads. We have assigned Traffic_Road_1 (PA6) for the

north-south roads as the traffic load and Traffic_Road_2 (PA8) for the east-west roads. Here we intended to include the functional part of the solved system which will determine which traffic signals to set and which to reset.

UARTx/USARTx Ports and Pins		
UART/USART	Tx	Rx
USART2	PA2	PA3
UART4	PA0	PA1
UART5	PC12	PD2

Here, we havee set up proper UARTx/USARTx ports for the communication vetween UART4-UAR5 and USART2-PC(Hercules). In the board, we have connected PA0-PD2 and PA1-PC12 for UART to UART communication. Because of this connection, the configuration command from Hercules to USART2 will be down to UART5,the monitor center via UART4, the control center.

6 Methodology:

Here, we have properly configured our GPIOx, USARTx/UARTx and TIMx. We have introduced functions necessary to get and send strings and characters over the communication bridge. Then we have parsed configuration command lines to manage traffic signal lines, monitoring the traffic and reading the road status. The parsed integer values for traffic signal light will be passed to UART5 via UART4 so that the logic behind automated traffic system is implemented.

When the road status or traffic monitor configuration time is given as command, the relevant information will passed as a string to UART4 via UART5 so that UART4 can pass the said string over to USART2 for PC communication.

We have included the configuration of peripherals in the report.We have properly documented our functions and variables in the zip file provided.