

Wyatt Tack  
Kalon Bienz  
EE 329-01 F'24  
Group H  
2024-Oct-25

## **EE 329 A7**

This code is designed to use the UART peripheral to communicate across USB to the targeted device, using a VT100 Terminal. The STM32 produces text that acts as a rudimentary game, displaying a splash screen with a bubbles graphic, and a boarder and character that move, and scroll to the next screen when jumped. This game was used with the terminal PuTTY in mind.

### **Link to YouTube Presentation:**

<https://youtu.be/cXt5tV1yKHU>

### **Calculations:**

Equation A7.a(a): Baud Rate From Reference Manual:

#### **41.4.4 LPUART baud rate generation**

The baud rate for the receiver and transmitter (Rx and Tx) are both set to the same value as programmed in the LPUART\_BRR register.

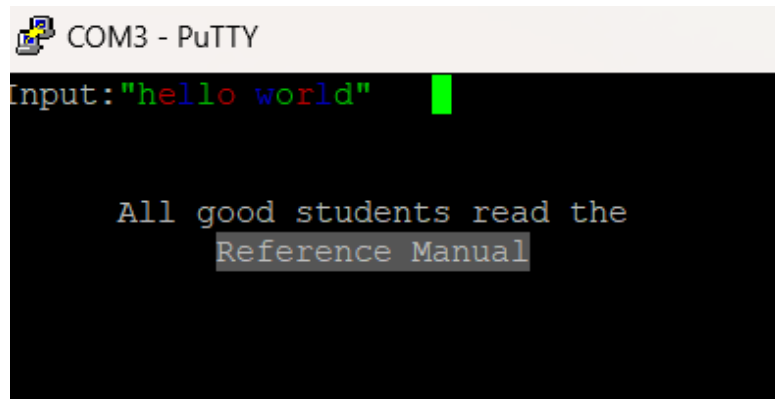
$$\text{Tx/Rx baud} = \frac{256 \times f_{CK}}{\text{LPUARTDIV}}$$

LPUARTDIV is coded on the LPUART\_BRR register.

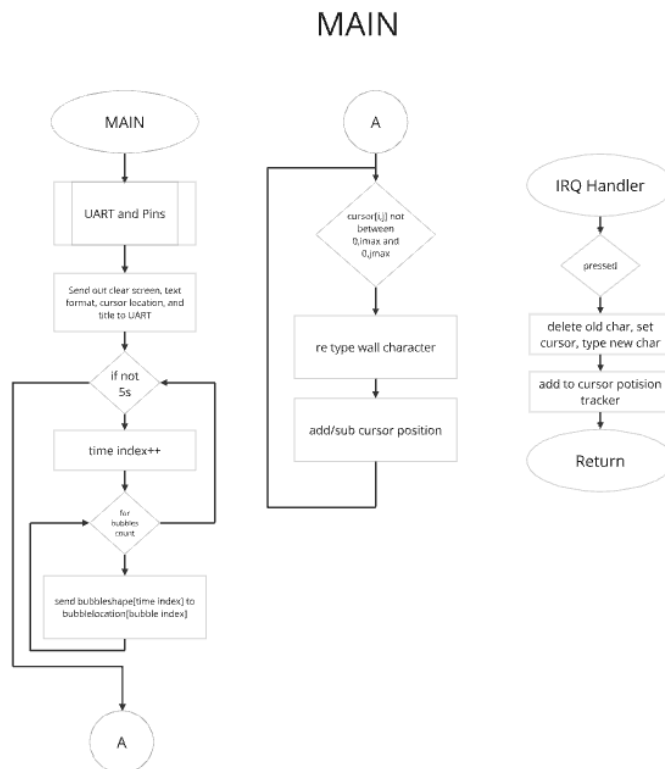
$$\text{LPUART} \rightarrow \text{DIV} = 256 \times 4\text{e}6 / 115200 = \underline{\underline{8889}}$$

## Captures:

Figure A7.a(a): Sample Home Screen



## Pseudocode Flow Chart:



## Formatted Source Code main.h:

```
/**
*****
* @file           : main.h
* project         : EE329 Lab A7
* author          : Wyatt Tack (wwt) - wtack@calpoly.edu
* date            : 10/12/2024
* firmware        : ST-Link V1
* @attention      : Copyright (c) 2024 STMicroelectronics. All rights
reserved.
*****
*
*      main header for defines for C and stm32 headers/hal
*
*****
*/

#ifndef __MAIN_H
#define __MAIN_H

#ifdef __cplusplus
extern "C" {
#endif

/* Created defines and function prototypes -----*/
void Execute(const char *messages[]);
void Bubbles(void);
void LPUART1_IRQHandler( void );

/* Includes -----*/
#include "stm32l4xx_hal.h"

/* Exported functions prototypes -----*/
void SystemClock_Config(void);
void Error_Handler(void);

#ifdef __cplusplus
}
#endif

#endif
```

## Formatted Source Code main.c:

## Formatted Source Code uart.h:

```
/**
*****
***
* @file          : uart.h
* project        : EE329 Lab A7
* author         : Wyatt Tack (wwt) - wtack@calpoly.edu
* date           : 10/13/2024
* firmware       : ST-Link V1
* @attention     : Copyright (c) 2024 STMicroelectronics. All rights
reserved.

*****
***
*
*   main header for defines for uart.h
*

*****
***
*/

#ifndef INC_UART_H_
#define INC_UART_H_

#include "stm32l4xx_hal.h"

#define BAUD_RATE (8889) //256 * 4MHz / 115.2kb/s = 8888.8

void UART_Init(void);
void LPUART_Print( const char* message );
//void LPUART1_IRQHandler( void );

#endif /* INC_UART_H_ */
```

## Formatted Source Code uart.c:

```
/**
 * *****
 * @file                : uart.c
 * project              : EE329 Lab A7
 * author               : Wyatt Tack (wvt) - wtack@calpoly.edu
 * date                 : 10/13/2024
 * firmware              : ST-Link V1
 * @attention           : Copyright (c) 2024 STMicroelectronics. All rights reserved.
 * *****
 *
 * Functions for UART module, set up as LPUART1 through GPIOG.
 * PTG-7 -> Tx
 * PTG-8 -> Rx
 *
 * *****
 */

#include "uart.h"

// -----
void UART_Init(void){

    //Power and Clock
    PWR->CR2 |= (PWR_CR2_IOSV); // power avail on PG[15:2] (LPUART1)
    RCC->AHB2ENR |= (RCC_AHB2ENR_GPIOGEN); // enable GPIOG clock
    RCC->APB1ENR2 |= RCC_APB1ENR2_LPUART1EN; // enable LPUART clock bridge
    //GPIO Ports - AF8, no PU/PD, fast (despite uart being slow)
    GPIOG->MODER &= ~(GPIO_MODER_MODE7 | GPIO_MODER_MODE8);
    GPIOG->MODER |= (GPIO_MODER_MODE7_1 | GPIO_MODER_MODE8_1);
    GPIOG->OTYPER &= ~(GPIO_OTYPER_OT7 | GPIO_OTYPER_OT8);
    GPIOG->PUPDR &= ~(GPIO_PUPDR_PUPD7 | GPIO_PUPDR_PUPD8);
    GPIOG->OSPEEDR |= ((3 << GPIO_OSPEEDR_OSPEED7_Pos) | (3 << GPIO_OSPEEDR_OSPEED8_Pos));
    GPIOG->AFR[0] &= ~(0x000F << GPIO_AFRL_AFSEL7_Pos);
    GPIOG->AFR[0] |= (0x0008 << GPIO_AFRL_AFSEL7_Pos);
    GPIOG->AFR[1] &= ~(0x000F << GPIO_AFRH_AFSEL8_Pos);
    GPIOG->AFR[1] |= (0x0008 << GPIO_AFRH_AFSEL8_Pos);
    //LPUART
    LPUART1->CR1 &= ~(USART_CR1_M1 | USART_CR1_M0); // 8-bit data
    LPUART1->CR1 |= USART_CR1_UE; // enable LPUART1
    LPUART1->CR1 |= (USART_CR1_TE | USART_CR1_RE); // enable xmit & rcv
    LPUART1->CR1 |= USART_CR1_RXNEIE; // enable LPUART1 rcv interrupt
    LPUART1->ISR &= ~(USART_ISR_RXNE); // clear Recv-Not-Empty flag
    LPUART1->BRR = (BAUD_RATE);
    /* USER: set baud rate register (LPUART1->BRR) */
    NVIC->ISER[2] = (1 << (LPUART1_IRQn & 0x1F)); // enable LPUART1 ISR
    __enable_irq();

}

void LPUART_Print( const char* message ) {
    uint16_t iStrIdx = 0;
    while ( message[iStrIdx] != 0 ) {
        while(!(LPUART1->ISR & USART_ISR_TXE)) // wait for empty xmit buffer
            ;
        LPUART1->TDR = message[iStrIdx]; // send this character
        iStrIdx++; // advance index to next char
    }
}

void LPUART_Print_Char (uint8_t charRecv){
    while( !(LPUART1->ISR & USART_ISR_TXE) );// wait for empty TX buffer
    LPUART1->TDR = charRecv; // send char to terminal
}
```