

УТВЕРЖДАЮ

Генеральный директор

_____ К.Н. Мигун
« ____ » _____ 2023 г.

УЗЕЛ ПЕЧАТНЫЙ R 6 LME 02 M

Встроенное программное обеспечение

Текст программы

РОФ.ГРЛМ.03001-01 12 01

Листов 16

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Перв. примен.	ГРЛМ.468223.002	АННОТАЦИЯ									
		<p>В данном документе содержится текст программы встроенного программного обеспечения узла печатного R 6 LME 02 M.</p> <p>В разделе «Необходимый комплект оборудования и ПО» указаны технические и программные средства, необходимые для просмотра электронной записи исходного кода встроенного программного обеспечения.</p> <p>В разделе «Характеристики записи текста программы» указаны характеристики исходного кода встроенного программного обеспечения, включая язык программирования, на котором написана программа, и место размещения электронной версии текста программы.</p> <p>В разделе «Фрагмент текста программы» приведен фрагмент исходного кода встроенного программного обеспечения узла печатного R 6 LME 02 M.</p>									
Справ. №											
Подп. и дата											
Инв. № дубл.											
Взам. инв. №											
Подп. и дата											
Инв. № подл.											

Файл	РОФ.ГРЛМ.03001-01 12 01.pdf				
Контрольная сумма					
0	Нов.				
Изм	Лист	№ докум.	Подп.	Дата	
Разраб.					
Пров.					
Н.контр.					
Утв.					

РОФ.ГРЛМ.03001-01 12 01					
Узел печатный R 6 LME 02 M. Встроенное программное обеспечение. Текст программы					
Лит.	Лист	Листов			
0	2	16			
ООО «Группа промышленных технологий»					

3 ФРАГМЕНТ ТЕКСТА ПРОГРАММЫ

- персональный компьютер с операционной системой Windows не ниже Windows 7;

- текстовый редактор для просмотра ASCII-текстовых файлов.

2 ХАРАКТЕРИСТИКИ ЗАПИСИ ТЕКСТА ПРОГРАММЫ

Текст программы включает в себя исходный код встроенного программного обеспечения узла печатного R 6 LME 02 M с комментариями, в которых указаны функциональное назначение представленных процедур.

Размещение файла: <http://gitlab.git-holding.ru:9071/git/meta-git>

3 ФРАГМЕНТ ТЕКСТА ПРОГРАММЫ

```

/* Private includes -----*/
/* USER CODE BEGIN Includes */
#include "app_ethernet.h"
#include "httpserver-netconn.h"
#include "dbg_uart.h"
#include "Base64.h"
#include "main_logic.h"
#include "global_vars.h"
#include "uart.h"
#include "main_logic.h"
/* USER CODE END Includes */

```

```

/* Private typedef ----- */
/* USER CODE BEGIN PTD */

/* USER CODE END PTD */

```

```

/* Private define -----*/
/* USER CODE BEGIN PD */
/* USER CODE END PD */

/* Private macro -----*/
/* USER CODE BEGIN PM */

/* USER CODE END PM */

/* Private variables -----*/
I2C_HandleTypeDef hi2c1;

osThreadId defaultTaskHandle;
osThreadId mainLogicTaskHandle;
uint32_t mainLogicTaskBuffer[ 512 ];
osStaticThreadDef_t mainLogicTaskControlBlock;
osTimerId timer1secHandle;
osStaticTimerDef_t timer1secControlBlock;
/* USER CODE BEGIN PV */
extern struct netif gnetif;
/* USER CODE END PV */

/* Private function prototypes -----*/
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_USART1_UART_Init(void);
static void MX_I2C1_Init(void);
static void MX_USART3_UART_Init(void);
void ThreadDefaultTask(void const * argument);
extern void ThreadMainLogic(void const * argument);
extern void tick1sec(void const * argument);

/* USER CODE BEGIN PFP */

/* USER CODE END PFP */

/* Private user code -----*/
/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

/**
 * @brief The application entry point.
 * @retval int
 */
int main(void)
{

```

Ине. № подл.	Подп. и дата	Ине. № дубл.	Подп. и дата	Взам. инв. №	Ине. № дубл.	Подп. и дата	Ине. № подл.
0	Нов.						
Изм.	Лист	№ докум.	Подп.	Дата			

```

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */

/* MCU Configuration-----*/

/* Reset of all peripherals, Initializes the Flash interface and the Systick. */
HAL_Init();

/* USER CODE BEGIN Init */

/* Reset the RCC clock configuration to the default reset state -----*/
/* Set HSION bit */
RCC->CR |= (uint32_t)0x00000001;

/* Reset CFGR register */
RCC->CFGR = 0x00000000;

/* Reset HSEON, CSSON and PLLON bits */
RCC->CR &= (uint32_t)0xFE6FFFFF;

/* Reset PLLCFGR register */
RCC->PLLCFGR = 0x24003010;

/* Reset HSEBYP bit */
RCC->CR &= (uint32_t)0xFFBFFFFF;

/* Disable all interrupts */
RCC->CIR = 0x00000000;
/* USER CODE END Init */

/* Configure the system clock */
SystemClock_Config();

/* USER CODE BEGIN SysInit */

/* USER CODE END SysInit */

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_USART1_UART_Init();
MX_I2C1_Init();
MX_USART3_UART_Init();
/* USER CODE BEGIN 2 */
    dbgUartResumeTx();
    dbgUartReceiveStart();
    uartInit();

```

Инв. № подл.	Подп. и дата				Лист 5
	Инв. № дубл.				
	Взам. инв. №				
	Подп. и дата				
0	Нов.				РОФ.ГРЛМ.03001-01 12 01
Изм.	Лист	№ докум.	Подп.	Дата	

initEepVars();

/* USER CODE END 2 */

/* USER CODE BEGIN RTOS_MUTEX */

/* add mutexes, ... */

/* USER CODE END RTOS_MUTEX */

/* USER CODE BEGIN RTOS_SEMAPHORES */

/* add semaphores, ... */

/* USER CODE END RTOS_SEMAPHORES */

/* Create the timer(s) */

/* definition and creation of timer1sec */

osTimerStaticDef(timer1sec, tick1sec, &timer1secControlBlock);

timer1secHandle = osTimerCreate(osTimer(timer1sec), osTimerPeriodic, NULL);

/* USER CODE BEGIN RTOS_TIMERS */

/* start timers, add new ones, ... */

/* USER CODE END RTOS_TIMERS */

/* USER CODE BEGIN RTOS_QUEUES */

/* add queues, ... */

/* USER CODE END RTOS_QUEUES */

/* Create the thread(s) */

/* definition and creation of defaultTask */

osThreadDef(defaultTask, ThreadDefaultTask, osPriorityNormal, 0, 1024);

defaultTaskHandle = osThreadCreate(osThread(defaultTask), NULL);

/* definition and creation of mainLogicTask */

osThreadStaticDef(mainLogicTask, ThreadMainLogic, osPriorityNormal, 0, 512,
mainLogicTaskBuffer, &mainLogicTaskControlBlock);

mainLogicTaskHandle = osThreadCreate(osThread(mainLogicTask), NULL);

/* USER CODE BEGIN RTOS_THREADS */

/* add threads, ... */

osTimerStart(timer1secHandle, 1000);

/* USER CODE END RTOS_THREADS */

/* Start scheduler */

osKernelStart();

/* We should never get here as control is now taken by the scheduler */

/* Infinite loop */

/* USER CODE BEGIN WHILE */

while (1)

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата						
0	Нов.				РОФ.ГРЛМ.03001-01 12 01					Лист
Изм.	Лист	№ докум.	Подп.	Дата						6

```

{
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
}
/* USER CODE END 3 */
}

/**
 * @brief System Clock Configuration
 * @retval None
 */
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};

    /** Configure the main internal regulator output voltage
    */
    __HAL_RCC_PWR_CLK_ENABLE();

    __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE1);

    /** Initializes the CPU, AHB and APB busses clocks
    */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
    RCC_OscInitStruct.HSEState = RCC_HSE_ON;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
    RCC_OscInitStruct.PLL.PLLM = 4;
    RCC_OscInitStruct.PLL.PLLN = 180;
    RCC_OscInitStruct.PLL.PLLP = RCC_PLLP_DIV2;
    RCC_OscInitStruct.PLL.PLLQ = 7;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }
    /** Activate the Over-Drive mode
    */
    if (HAL_PWREx_EnableOverDrive() != HAL_OK)
    {
        Error_Handler();
    }
    /** Initializes the CPU, AHB and APB busses clocks
    */
    RCC_ClkInitStruct.ClockType
    RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYCLK

```

Ине. № подл.	Подп. и дата	Ине. № дубл.	Взам. инв. №	Подп. и дата
0	Нов.			
Изм.	Лист	№ докум.	Подп.	Дата

```

        |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV4;
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV2;

if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_5) !=
HAL_OK)
{
    Error_Handler();
}

/**
 * @brief I2C1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_I2C1_Init(void)
{

    /* USER CODE BEGIN I2C1_Init 0 */

    /* USER CODE END I2C1_Init 0 */

    /* USER CODE BEGIN I2C1_Init 1 */

    /* USER CODE END I2C1_Init 1 */
    hi2c1.Instance = I2C1;
    hi2c1.Init.ClockSpeed = 400000;
    hi2c1.Init.DutyCycle = I2C_DUTYCYCLE_2;
    hi2c1.Init.OwnAddress1 = 0;
    hi2c1.Init.AddressingMode = I2C_ADDRESSINGMODE_7BIT;
    hi2c1.Init.DualAddressMode = I2C_DUALADDRESS_DISABLE;
    hi2c1.Init.OwnAddress2 = 0;
    hi2c1.Init.GeneralCallMode = I2C_GENERALCALL_DISABLE;
    hi2c1.Init.NoStretchMode = I2C_NOSTRETCH_DISABLE;
    if (HAL_I2C_Init(&hi2c1) != HAL_OK)
    {
        Error_Handler();
    }
    /** Configure Analogue filter
     */
    if (HAL_I2CEx_ConfigAnalogFilter(&hi2c1,
I2C_ANALOGFILTER_ENABLE) != HAL_OK)
    {
        Error_Handler();
    }

```

Ине. № подл.	Подп. и дата				Лист
	Ине. № дубл.				
	Взам. инв. №				
	Подп. и дата				
0	Нов.				РОФ.ГРЛМ.03001-01 12 01 8
Изм.	Лист	№ докум.	Подп.	Дата	


```

}
/** Configure Digital filter
*/
if (HAL_I2CEx_ConfigDigitalFilter(&hi2c1, 0) != HAL_OK)
{
    Error_Handler();
}
/* USER CODE BEGIN I2C1_Init 2 */

/* USER CODE END I2C1_Init 2 */

}

/**
 * @brief USART1 Initialization Function
 * @param None
 * @retval None
 */
static void MX_USART1_UART_Init(void)
{
    /* USER CODE BEGIN USART1_Init 0 */

    /* USER CODE END USART1_Init 0 */

    LL_USART_InitTypeDef USART_InitStruct = {0};

    LL_GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* Peripheral clock enable */
    LL_APB2_GRP1_EnableClock(LL_APB2_GRP1_PERIPH_USART1);

    LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOB);
    /**USART1 GPIO Configuration
    PB6 -----> USART1_TX
    PB7 -----> USART1_RX
    */
    GPIO_InitStruct.Pin = LL_GPIO_PIN_6|LL_GPIO_PIN_7;
    GPIO_InitStruct.Mode = LL_GPIO_MODE_ALTERNATE;
    GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_VERY_HIGH;
    GPIO_InitStruct.OutputType = LL_GPIO_OUTPUT_PUSHPULL;
    GPIO_InitStruct.Pull = LL_GPIO_PULL_NO;
    GPIO_InitStruct.Alternate = LL_GPIO_AF_7;
    LL_GPIO_Init(GPIOB, &GPIO_InitStruct);

    /* USART1 interrupt Init */

```

Инв. № подл.	Подп. и дата				Лист 9
	Инв. № дубл.				
	Взам. инв. №				
	Подп. и дата				
0	Нов.				РОФ.ГРЛМ.03001-01 12 01
Изм.	Лист	№ докум.	Подп.	Дата	

```

NVIC_SetPriority(USART1_IRQn,
EncodePriority(NVIC_GetPriorityGrouping(),15, 0));
NVIC_EnableIRQ(USART1_IRQn);

```

```
/* USER CODE BEGIN USART1_Init 1 */
```

```
/* USER CODE END USART1 Init 1 */
```

```
USART_InitStruct.BaudRate = 115200;
```

```
USART_InitStruct.DataWidth = LL_USART_DATAWIDTH_8B;
```

```
USART_InitStruct.StopBits = LL_USART_STOPBITS_1;
```

```
USART_InitStruct.Parity = LL_USART_PARITY_NONE;
```

```
USART_InitStruct.TransferDirection = LL_USART_DIRECTION_TX_RX;
```

USART InitStruct.HardwareFlowControl

```
LL USART HWCONTROL NONE;
```

```
USART_InitStructure.OverSampling = LL_USART_OVERSAMPLING_16;
```

```
LL_USART_Init(USART1, &USART_InitStruct);
```

```
LL_USART_ConfigAsyncMode(USART1);
```

```
LL_USART_Enable(USART1);
```

```
/* USER CODE BEGIN USART1_Init 2 */
```

```
/* USER CODE END USART1 Init 2 */
```

}

/**

* @brief USART3 Initialization Function

* @param None

* @retval None

*/

```
static void MX_USART3_UART_Init(void)
```

 $\{$

```
/* USER CODE BEGIN USART3_Init 0 */
```

```
/* USER CODE END USART3 Init 0 */
```

```
LL_USART_InitTypeDef USART_InitStructure = {0};
```

```
LL_GPIO_InitTypeDef GPIO_InitStructure = {0};
```

```
/* Peripheral clock enable */
```

```
LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_USART3);
```

```
LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOB);
```

/**USART3 GPIO Configuration

PB10 -----> USART3_TX

PB11 -----> USART3_RX

Инв. № подл.	Подп. и дата					<pre> /** * @brief USART3 Initialization Function * @param None * @retval None */ static void MX_USART3_UART_Init(void) { /* USER CODE BEGIN USART3_Init 0 */ /* USER CODE END USART3_Init 0 */ LL_USART_InitTypeDef USART_InitStruct = {0}; LL_GPIO_InitTypeDef GPIO_InitStruct = {0}; /* Peripheral clock enable */ LL_APB1_GRP1_EnableClock(LL_APB1_GRP1_PERIPH_USART3); LL_AHB1_GRP1_EnableClock(LL_AHB1_GRP1_PERIPH_GPIOB); /**USART3 GPIO Configuration PB10 -----> USART3_TX PB11 -----> USART3_RX </pre>
Инв. № дубл.	Подп. и дата					
Взам. инв. №	Подп. и дата					
Инв. № подл.	Подп. и дата					
Инв. № подл.	Подп. и дата					<p style="text-align: center;">РОФ.ГРЛМ.03001-01 12 01</p>

```

*/
GPIO_InitStruct.Pin = LL_GPIO_PIN_10|LL_GPIO_PIN_11;
GPIO_InitStruct.Mode = LL_GPIO_MODE_ALTERNATE;
GPIO_InitStruct.Speed = LL_GPIO_SPEED_FREQ_VERY_HIGH;
GPIO_InitStruct.OutputType = LL_GPIO_OUTPUT_PUSHPULL;
GPIO_InitStruct.Pull = LL_GPIO_PULL_NO;
GPIO_InitStruct.Alternate = LL_GPIO_AF_7;
LL_GPIO_Init(GPIOB, &GPIO_InitStruct);

/* USART3 interrupt Init */
NVIC_SetPriority(USART3_IRQn,
NVIC_EncodePriority(NVIC_GetPriorityGrouping(),6, 0));
NVIC_EnableIRQ(USART3_IRQn);

/* USER CODE BEGIN USART3_Init 1 */

/* USER CODE END USART3_Init 1 */
USART_InitStruct.BaudRate = 115200;
USART_InitStruct.DataWidth = LL_USART_DATAWIDTH_8B;
USART_InitStruct.StopBits = LL_USART_STOPBITS_1;
USART_InitStruct.Parity = LL_USART_PARITY_NONE;
USART_InitStruct.TransferDirection = LL_USART_DIRECTION_TX_RX;
USART_InitStruct.HardwareFlowControl =
LL_USART_HWCONTROL_NONE;
USART_InitStruct.OverSampling = LL_USART_OVERSAMPLING_16;
LL_USART_Init(USART3, &USART_InitStruct);
LL_USART_ConfigAsyncMode(USART3);
LL_USART_Enable(USART3);
/* USER CODE BEGIN USART3_Init 2 */

/* USER CODE END USART3_Init 2 */

}

/**
 * @brief GPIO Initialization Function
 * @param None
 * @retval None
 */
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStruct = {0};

    /* GPIO Ports Clock Enable */
    __HAL_RCC_GPIOE_CLK_ENABLE();
    __HAL_RCC_GPIOF_CLK_ENABLE();
    __HAL_RCC_GPIOH_CLK_ENABLE();

```

Ине. № подл.	Подп. и дата	Ине. № дубл.	Взам. инв. №	Подп. и дата	Ине. № подл.
0	Нов.				
Изм.	Лист	№ докум.	Подп.	Дата	

```

    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
    __HAL_RCC_GPIOG_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOE,
LED_E1_Pin|LED_E2_Pin|LED_E3_Pin|LED_E4_Pin
                        |LED_E5_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOF,                                LED_E6_Pin|LED_ERR_Pin,
GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(LED_ON_GPIO_Port, LED_ON_Pin, GPIO_PIN_SET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(GPIOB, LD1_Pin|RTS3_NAP_Pin, GPIO_PIN_RESET);

    /*Configure GPIO pin Output Level */
    HAL_GPIO_WritePin(LED_0_GPIO_Port, LED_0_Pin, GPIO_PIN_SET);

    /*Configure GPIO pins : LED_E1_Pin LED_E2_Pin LED_E3_Pin LED_E4_Pin
                        LED_E5_Pin */
    GPIO_InitStruct.Pin = LED_E1_Pin|LED_E2_Pin|LED_E3_Pin|LED_E4_Pin
                        |LED_E5_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOE, &GPIO_InitStruct);

    /*Configure GPIO pins : LED_E6_Pin LED_ERR_Pin LED_ON_Pin */
    GPIO_InitStruct.Pin = LED_E6_Pin|LED_ERR_Pin|LED_ON_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOF, &GPIO_InitStruct);

    /*Configure GPIO pin : LD1_Pin */
    GPIO_InitStruct.Pin = LD1_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_NOPULL;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(LD1_GPIO_Port, &GPIO_InitStruct);

```

Инв. № подл.	Подп. и дата				Лист
	Инв. № дубл.				
	Взам. инв. №				
	Подп. и дата				
<div>/*Configure GPIO pins : LED_E1_Pin LED_E2_Pin LED_E3_Pin LED_E4_Pin LED_E5_Pin */ GPIO_InitStruct.Pin = LED_E1_Pin LED_E2_Pin LED_E3_Pin LED_E4_Pin LED_E5_Pin; GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP; GPIO_InitStruct.Pull = GPIO_NOPULL; GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW; HAL_GPIO_Init(GPIOE, &GPIO_InitStruct); /*Configure GPIO pins : LED_E6_Pin LED_ERR_Pin LED_ON_Pin */ GPIO_InitStruct.Pin = LED_E6_Pin LED_ERR_Pin LED_ON_Pin; GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP; GPIO_InitStruct.Pull = GPIO_NOPULL; GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW; HAL_GPIO_Init(GPIOF, &GPIO_InitStruct); /*Configure GPIO pin : LD1_Pin */ GPIO_InitStruct.Pin = LD1_Pin; GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP; GPIO_InitStruct.Pull = GPIO_NOPULL; GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW; HAL_GPIO_Init(LD1_GPIO_Port, &GPIO_InitStruct);</div>					
0	Нов.				РОФ.ГРЛМ.03001-01 12 01
Изм.	Лист	№ докум.	Подп.	Дата	
					12

```

/*Configure GPIO pin : START_Pin */
GPIO_InitStruct.Pin = START_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(START_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pin : RTS3_NAP_Pin */
GPIO_InitStruct.Pin = RTS3_NAP_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
HAL_GPIO_Init(RTS3_NAP_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pin : LED_0_Pin */
GPIO_InitStruct.Pin = LED_0_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
GPIO_InitStruct.Pull = GPIO_NOPULL;
GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
HAL_GPIO_Init(LED_0_GPIO_Port, &GPIO_InitStruct);

/*Configure GPIO pins : PS1_Pin PS2_Pin PS3_Pin PS4_Pin
                        PS5_Pin PS6_Pin */
GPIO_InitStruct.Pin = PS1_Pin|PS2_Pin|PS3_Pin|PS4_Pin
                        |PS5_Pin|PS6_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_NOPULL;
HAL_GPIO_Init(GPIOD, &GPIO_InitStruct);

/*Configure GPIO pin : BUT_Pin */
GPIO_InitStruct.Pin = BUT_Pin;
GPIO_InitStruct.Mode = GPIO_MODE_INPUT;
GPIO_InitStruct.Pull = GPIO_PULLUP;
HAL_GPIO_Init(BUT_GPIO_Port, &GPIO_InitStruct);

}

/* USER CODE BEGIN 4 */

/* USER CODE END 4 */

/* USER CODE BEGIN Header_ThreadDefaultTask */
/**
 * @brief Function implementing the defaultTask thread.
 * @param argument: Not used
 * @retval None
 */
/* USER CODE END Header_ThreadDefaultTask */

```

Ине. № подл.	Подп. и дата	Взам. инв. №	Ине. № дубл.	Подп. и дата

```

void ThreadDefaultTask(void const * argument)
{
    /* init code for LWIP */
    MX_LWIP_Init();
    /* USER CODE BEGIN 5 */

    /* Initialize webserver demo */
    http_server_netconn_init();
    dbgUartPuts("Start ThreadDefaultTask.\r");
    /* Infinite loop */
    for(;;)
    {
        defaultTaskMainLoop();
    }
    /* USER CODE END 5 */
}

/**
 * @brief Period elapsed callback in non blocking mode
 * @note This function is called when TIM1 interrupt took place, inside
 * HAL_TIM_IRQHandler(). It makes a direct call to HAL_IncTick() to increment
 * a global variable "uwTick" used as application time base.
 * @param htim : TIM handle
 * @retval None
 */
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    /* USER CODE BEGIN Callback 0 */

    /* USER CODE END Callback 0 */
    if (htim->Instance == TIM1) {
        HAL_IncTick();
    }
    /* USER CODE BEGIN Callback 1 */

    /* USER CODE END Callback 1 */
}

/**
 * @brief This function is executed in case of error occurrence.
 * @retval None
 */
void Error_Handler(void)
{
    /* USER CODE BEGIN Error_Handler_Debug */
    /* User can add his own implementation to report the HAL error return state */

```

Инв. № подл.	Подп. и дата				Лист
	Инв. № дубл.				
	Взам. инв. №				
Инв. № подл.	Подп. и дата				Лист
	Инв. № дубл.				
	Взам. инв. №				
0	Нов.				РОФ.ГРЛМ.03001-01 12 01
Изм.	Лист	№ докум.	Подп.	Дата	

```

/* USER CODE END Error_Handler_Debug */
}

#ifdef USE_FULL_ASSERT
/**
 * @brief Reports the name of the source file and the source line number
 *        where the assert_param error has occurred.
 * @param file: pointer to the source file name
 * @param line: assert_param error line source number
 * @retval None
 */
void assert_failed(uint8_t *file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
    tex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}
#endif /* USE_FULL_ASSERT */

```

Инв. № подл.	Подп. и дата				Инв. № дубл.	Подп. и дата				Взам. инв. №	Инв. № подл.	Лист
0	Нов.				РОФ.ГРЛМ.03001-01 12 01						15	
Изм.	Лист	№ докум.	Подп.	Дата								

ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ

[illegible][illegible]