

Министерство образования Республики Беларусь
Учреждение образования
**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ**

Факультет компьютерных систем и сетей
Кафедра программного обеспечения информационных технологий

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ВСТРОЕННЫХ СИСТЕМ

ОТЧЁТ

По лабораторной работе №4

Тема работы:

Использование Bluetooth LE модуля HM-10 с STM-32

Выполнили:

Шуляк А.В.
Дранкевич А.А.

Проверил:

Леванцевич В. А.

Минск 2023

1. Чип HM-10

Модуль HM-10 представляет собой Bluetooth-модуль, который широко используется для беспроводной связи в различных электронных устройствах. Основные характеристики и особенности чипа HM-10:

- Работа в режимах Master и Slave
- Поддержка U(S)ART
- Питание - 5 (3.3) В
- Потребление модуля - 8.5 мА в активном состоянии
- Поддержка Bluetooth Low Energy
- Поддержка AT-команд для управления

2. Настройка оборудования

Перед подключением модуля к STM32 его необходимо сбросить до заводских настроек для обеспечения гарантии его корректной работы. Для этого необходимо подключить модуль через USB-переходник, подключиться к соответствующему последовательному виртуальному порту и отправить команду «AT+RENEW\r\n», ожидаемый ответ модуля - «OK+RENEW\r\n».

Для непосредственного подключения модуля к STM32 необходимо подключить пины модуля к плате следующим образом:

- VDD к пину +3.3V (+5V) STM32
- GND к соответствующему пину STM32
- RX к пину PA2
- TX к пину PA10

После чего необходимо настроить USART и DMA для дуплексной передачи данных между микроконтроллером и модулем HM-10, как это указано в Приложении А, файл с именем Lab4.ioc

3. Работа модуля

Для работы модуля HM-10 его необходимо корректно настроить, для этого с STM32 нужно отправить по интерфейсу USART на модуль следующие команды:

1. TRANSMIT(huart1, "AT+MODE0\r\n"); – проверка работоспособности модуля, ожидаемый ответ - "OK\r\n"

2. TRANSMIT(huart1, "AT+MODE0\r\n"); – установка режима работы модуля – в 0-ом режиме модуль реагирует на управляющие команды только без наличия подключённого к нему по Bluetooth узла, при подключении переходит в режим пересылки данных
3. TRANSMIT(huart1, "AT+NAMESTUPID_STM\r\n"); – установка имени для Bluetooth-маячка, под которым он будет обнаружен клиентами
4. TRANSMIT(huart1, "AT+RESET\r\n"); – перезагрузка модуля, необходима для того, чтобы все внесённые изменения в настройках вступили в силу

После выполнения этих шагов модуль начинает корректно реагировать на попытки подключения (устройство с Android и приложением BLE Terminal). В рамках лабораторной работы был реализован «эхо-сервер», который ожидает от клиента отправки 4-символьной строковой последовательности, а затем отправляет её клиенту назад. Пересылка строк произвольной длины затруднена, поскольку данный модуль не предназначен для отправки большого количества данных между подключёнными узлами, а служит именно для обмена короткими сообщениями.

ПРИЛОЖЕНИЕ А

Исходный код программы

Main.c

```
#include "main.h"
#include <stdio.h>
#include <string.h>
#define true (bool)1;
#define false (bool)0;
#define UART_TIMEOUT 1000
#define CHECK_ON(what) (strcmp((char *)buffer, (what), sizeof (what)) == 0)
#define TRANSMIT(uart, msg) err = HAL_UART_Transmit(&(uart), (unsigned char *) (msg),
sizeof (msg) - 1, UART_TIMEOUT);\
                                if (err != HAL_OK) {\
                                    while (1) {\
                                        HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET); HAL_Delay(100);\
                                        HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET); HAL_Delay(100);\
                                        }\
                                    }
                                int myreceive(UART_HandleTypeDef *uart, unsigned char *buffer, int required) {
                                    int i = 0;
                                    while ((HAL_UART_Receive(uart, buffer + i, 1, UART_TIMEOUT) == HAL_OK) && (i <
required)) {i++; };
                                    return i;
                                }
                                #define RECIEVE(uart, msg) if ((myreceive(&(uart), buffer, 256) == 0) || !CHECK_ON((msg)))
{\
                                while (1) {\
                                    HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET); HAL_Delay(100);\
                                    HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET); HAL_Delay(100);\
                                    }\
                                }
                                #define RECIEVEN(uart, num) if (myreceive(&(uart), buffer, num) != num) {\
                                    while (1) {\
                                        HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_RESET); HAL_Delay(100);\
                                        HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET); HAL_Delay(100);\
                                        }\
                                    }
                                }

UART_HandleTypeDef huart1;
UART_HandleTypeDef huart2;
DMA_HandleTypeDef hdma_usart1_rx;
void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_DMA_Init(void);
static void MX_USART2_UART_Init(void);
static void MX_USART1_UART_Init(void);
volatile int listen_mode = 0;
volatile int bpos = 0;
unsigned char buffer[256];
unsigned char buffer_a[256];
unsigned char buffer_b[256];
unsigned char *pb_a = buffer_a;
unsigned char *pb_b = buffer_b;
void HAL_UART_RxHalfCpltCallback(UART_HandleTypeDef *huart) {}
```

```

void HAL_UART_RxCpltCallback(UART_HandleTypeDef *huart)
{
    HAL_GPIO_TogglePin (LD2_GPIO_Port, LD2_Pin); // toggle PA0
    bpos=2;
    //HAL_UART_Transmit(&(huart1), (unsigned char *) (buffer), 6, UART_TIMEOUT);
    HAL_UART_Transmit(&(huart2), (unsigned char *) (buffer), 6, UART_TIMEOUT);
    if (listen_mode) {
        HAL_UART_Receive_DMA(&huart1, buffer, 6);
    }
}

int main(void)
{
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_DMA_Init();
    MX_USART2_UART_Init();
    MX_USART1_UART_Init();
    HAL_GPIO_WritePin(LD2_GPIO_Port, LD2_Pin, GPIO_PIN_SET);
    HAL_StatusTypeDef err;
    TRANSMIT(huart2, "AT\r\n");
    TRANSMIT(huart1, "AT\r\n");
    bpos = 0; // Clear receive flag
    err = HAL_UART_Receive_DMA (&huart1, buffer, sizeof ("OK\r\n") - 1); // request data
    while (bpos != 2) {};
    TRANSMIT(huart2, "AT+MODE0\r\n");
    TRANSMIT(huart1, "AT+MODE0\r\n");
    bpos = 0; // Clear receive flag
    err = HAL_UART_Receive_DMA (&huart1, buffer, sizeof ("OK+SET:0\r\n") - 1); // request
data
    while (bpos != 2) {};
    TRANSMIT(huart2, "AT+NAMESTUPID_STM\r\n");
    TRANSMIT(huart1, "AT+NAMESTUPID_STM\r\n");
    bpos = 0;
    err = HAL_UART_Receive_DMA (&huart1, buffer, sizeof ("OK+SET:STUPID_STM\r\n") - 1);
    while (bpos != 2) {};
    TRANSMIT(huart2, "AT+NAMEI_HATE_STM\r\n");
    TRANSMIT(huart1, "AT+NAMEI_HATE_STM\r\n");
    bpos = 0;
    err = HAL_UART_Receive_DMA (&huart1, buffer, sizeof ("OK+SET:I_HATE_STM\r\n") - 1);
    while (bpos != 2) {};
    TRANSMIT(huart2, "AT+RESET\r\n");
    TRANSMIT(huart1, "AT+RESET\r\n");
    bpos = 0;
    err = HAL_UART_Receive_DMA (&huart1, buffer, sizeof ("OK+RESET\r\n") - 1);
    while (bpos != 2) {};
    listen_mode = 1;
    HAL_UART_Receive_DMA(&huart1, buffer, 6);
    while (1) { };
    Error_Handler();
    TRANSMIT(huart2, "AT\r\n");
    TRANSMIT(huart1, "AT\r\n");
    HAL_Delay(500);
    RECIEVE(huart1, "OK");
    TRANSMIT(huart2, "AT+MODE0\r\n");
    TRANSMIT(huart1, "AT+MODE0\r\n");
    HAL_Delay(500);
    RECIEVE(huart1, "OK+SET:0");
    TRANSMIT(huart2, "AT+NAMESTUPID_STM\r\n");
    TRANSMIT(huart1, "AT+NAMESTUPID_STM\r\n");
    HAL_Delay(500);
    RECIEVE(huart1, "OK+SET:STUPID_STM");
    TRANSMIT(huart2, "AT+RESET\r\n");
    TRANSMIT(huart1, "AT+RESET\r\n");

```

```

    HAL_Delay(500);
    RECIEVE(huart1, "OK+RESET");
    while (1) {}
}
void SystemClock_Config(void)
{
    RCC_OscInitTypeDef RCC_OscInitStruct = {0};
    RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSISState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI_DIV2;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL16;
    if (HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
    {
        Error_Handler();
    }
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
                                |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
    if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_2) != HAL_OK)
    {
        Error_Handler();
    }
}
static void MX_USART1_UART_Init(void)
{
    huart1.Instance = USART1;
    huart1.Init.BaudRate = 9600;
    huart1.Init.WordLength = UART_WORDLENGTH_8B;
    huart1.Init.StopBits = UART_STOPBITS_1;
    huart1.Init.Parity = UART_PARITY_NONE;
    huart1.Init.Mode = UART_MODE_TX_RX;
    huart1.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart1.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart1) != HAL_OK)
    {
        Error_Handler();
    }
}
static void MX_USART2_UART_Init(void)
{
    huart2.Instance = USART2;
    huart2.Init.BaudRate = 9600;
    huart2.Init.WordLength = UART_WORDLENGTH_8B;
    huart2.Init.StopBits = UART_STOPBITS_1;
    huart2.Init.Parity = UART_PARITY_NONE;
    huart2.Init.Mode = UART_MODE_TX_RX;
    huart2.Init.HwFlowCtl = UART_HWCONTROL_NONE;
    huart2.Init.OverSampling = UART_OVERSAMPLING_16;
    if (HAL_UART_Init(&huart2) != HAL_OK)
    {
        Error_Handler();
    }
}
static void MX_DMA_Init(void)
{
    __HAL_RCC_DMA1_CLK_ENABLE();
    HAL_NVIC_SetPriority(DMA1_Channel5_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(DMA1_Channel5_IRQn);
}

```

```

}
static void MX_GPIO_Init(void)
{
    GPIO_InitTypeDef GPIO_InitStructure = {0};
    __HAL_RCC_GPIOC_CLK_ENABLE();
    __HAL_RCC_GPIOD_CLK_ENABLE();
    __HAL_RCC_GPIOA_CLK_ENABLE();
    __HAL_RCC_GPIOB_CLK_ENABLE();
    HAL_GPIO_WritePin(GPIOA, LED_Pin|LD2_Pin, GPIO_PIN_RESET);
    GPIO_InitStructure.Pin = B1_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_IT_RISING;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(B1_GPIO_Port, &GPIO_InitStructure);
    GPIO_InitStructure.Pin = BT_Status_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_INPUT;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    HAL_GPIO_Init(BT_Status_GPIO_Port, &GPIO_InitStructure);
    GPIO_InitStructure.Pin = LED_Pin|LD2_Pin;
    GPIO_InitStructure.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStructure.Pull = GPIO_NOPULL;
    GPIO_InitStructure.Speed = GPIO_SPEED_FREQ_LOW;
    HAL_GPIO_Init(GPIOA, &GPIO_InitStructure);
    HAL_NVIC_SetPriority(EXTI15_10_IRQn, 0, 0);
    HAL_NVIC_EnableIRQ(EXTI15_10_IRQn);
}
void Error_Handler(void)
{
    __disable_irq();
    while (1) { }
}

```

Lab4.ioc

```

#MicroXplorer Configuration settings - do not
modify
CAD.formats=
CAD.pinconfig=
CAD.provider=
Dma.Request0=USART1_RX
Dma.RequestsNb=1
Dma.USART1_RX.0.Direction=DMA_PERIPH_TO_MEMOR
Y
Dma.USART1_RX.0.Instance=DMA1_Channel5
Dma.USART1_RX.0.MemDataAlignment=DMA_MDATAALI
GN_BYTE
Dma.USART1_RX.0.MemInc=DMA_MINC_ENABLE
Dma.USART1_RX.0.Mode=DMA_CIRCULAR
Dma.USART1_RX.0.PeriphDataAlignment=DMA_PDATA
ALIGN_BYTE
Dma.USART1_RX.0.PeriphInc=DMA_PINC_DISABLE
Dma.USART1_RX.0.Priority=DMA_PRIORITY_VERY_HI
GH
Dma.USART1_RX.0.RequestParameters=Instance,Di
rection,PeriphInc,MemInc,PeriphDataAlignment,
MemDataAlignment,Mode,Priority
File.Version=6
GPIO.groupedBy=Group By Peripherals
KeepUserPlacement=false
Mcu.CPN=STM32F103RBT6
Mcu.Family=STM32F1
Mcu.IP0=DMA
Mcu.IP1=NVIC
Mcu.IP2=RCC

```

```

Mcu.IP3=SYS
Mcu.IP4=USART1
Mcu.IP5=USART2
Mcu.IPNb=6
Mcu.Name=STM32F103R(8-B)Tx
Mcu.Package=LQFP64
Mcu.Pin0=PC13-TAMPER-RTC
Mcu.Pin1=PC14-OSC32_IN
Mcu.Pin10=PA9
Mcu.Pin11=PA10
Mcu.Pin12=PA13
Mcu.Pin13=PA14
Mcu.Pin14=PB3
Mcu.Pin15=VP_SYS_VS_Systick
Mcu.Pin2=PC15-OSC32_OUT
Mcu.Pin3=PD0-OSC_IN
Mcu.Pin4=PD1-OSC_OUT
Mcu.Pin5=PA1
Mcu.Pin6=PA2
Mcu.Pin7=PA3
Mcu.Pin8=PA4
Mcu.Pin9=PA5
Mcu.PinsNb=16
Mcu.ThirdPartyNb=0
Mcu.UserConstants=
Mcu.UserName=STM32F103RBTx
MxCube.Version=6.9.1
MxDb.Version=DB.6.0.91
NVIC.BusFault_IRQn=true\0\0:false:false\
true:true:false:false

```

```

NVIC.DMA1_Channel5_IRQn=true\0\0:false\false\true\true\false\true\true
NVIC.DebugMonitor_IRQn=true\0\0:false\false\true\true\false\false
NVIC.EXTI15_10_IRQn=true\0\0:false\false\true\true\true\true
NVIC.ForceEnabledDMAVector=true
NVIC.HardFault_IRQn=true\0\0:false\false\true\true\false\false
NVIC.MemoryManagement_IRQn=true\0\0:false\false\true\true\false\false
NVIC.NonMaskableInt_IRQn=true\0\0:false\false\true\true\false\false
NVIC.PendSV_IRQn=true\0\0:false\false\true\true\false\false
NVIC.PriorityGroup=NVIC_PRIORITYGROUP_4
NVIC.SVCall_IRQn=true\0\0:false\false\true\true\false\false
NVIC.SysTick_IRQn=true\0\0:false\false\true\true\true\false
NVIC.USART1_IRQn=true\0\0:false\false\true\true\true\true
NVIC.UsageFault_IRQn=true\0\0:false\false\true\true\false\false
PA1.GPIOParameters=GPIO_Label
PA1.GPIO_Label=BT_Status
PA1.Locked=true
PA1.Signal=GPIO_Input
PA10.GPIOParameters=GPIO_Mode
PA10.GPIO_Mode=GPIO_MODE_INPUT
PA10.Mode=Asynchronous
PA10.Signal=USART1_RX
PA13.GPIOParameters=GPIO_Label
PA13.GPIO_Label=TMS
PA13.Locked=true
PA13.Mode=Serial_Wire
PA13.Signal=SYS_JTMS-SWDIO
PA14.GPIOParameters=GPIO_Label
PA14.GPIO_Label=TCK
PA14.Locked=true
PA14.Mode=Serial_Wire
PA14.Signal=SYS_JTCK-SWCLK
PA2.GPIOParameters=GPIO_Speed,GPIO_PuPd,GPIO_Label,GPIO_Mode
PA2.GPIO_Label=USART_TX
PA2.GPIO_Mode=GPIO_MODE_AF_PP
PA2.GPIO_PuPd=GPIO_NOPULL
PA2.GPIO_Speed=GPIO_SPEED_FREQ_LOW
PA2.Locked=true
PA2.Mode=Asynchronous
PA2.Signal=USART2_TX
PA3.GPIOParameters=GPIO_Speed,GPIO_PuPd,GPIO_Label,GPIO_Mode
PA3.GPIO_Label=USART_RX
PA3.GPIO_Mode=GPIO_MODE_AF_PP
PA3.GPIO_PuPd=GPIO_NOPULL
PA3.GPIO_Speed=GPIO_SPEED_FREQ_LOW
PA3.Locked=true
PA3.Mode=Asynchronous
PA3.Signal=USART2_RX
PA4.GPIOParameters=GPIO_Label
PA4.GPIO_Label=LED
PA4.Locked=true
PA4.Signal=GPIO_Output

```

```

PA5.GPIOParameters=GPIO_Speed,GPIO_PuPd,GPIO_Label,GPIO_ModeDefaultOutputPP
PA5.GPIO_Label=LD2 [Green Led]
PA5.GPIO_ModeDefaultOutputPP=GPIO_MODE_OUTPUT_PP
PA5.GPIO_PuPd=GPIO_NOPULL
PA5.GPIO_Speed=GPIO_SPEED_FREQ_LOW
PA5.Locked=true
PA5.Signal=GPIO_Output
PA9.Mode=Asynchronous
PA9.Signal=USART1_TX
PB3.GPIOParameters=GPIO_Label
PB3.GPIO_Label=SWO
PB3.Locked=true
PB3.Signal=SYS_JTDO-TRACESWO
PC13-TAMPER-
RTC.GPIOParameters=GPIO_PuPd,GPIO_Label
PC13-TAMPER-RTC.GPIO_Label=B1 [Blue PushButton]
PC13-TAMPER-RTC.GPIO_PuPd=GPIO_NOPULL
PC13-TAMPER-RTC.Locked=true
PC13-TAMPER-RTC.Signal=GPXTI13
PC14-OSC32_IN.Locked=true
PC14-OSC32_IN.Mode=LSE-External-Oscillator
PC14-OSC32_IN.Signal=RCC_OSC32_IN
PC15-OSC32_OUT.Locked=true
PC15-OSC32_OUT.Mode=LSE-External-Oscillator
PC15-OSC32_OUT.Signal=RCC_OSC32_OUT
PD0-OSC_IN.Locked=true
PD0-OSC_IN.Mode=HSE-External-Clock-Source
PD0-OSC_IN.Signal=RCC_OSC_IN
PD1-OSC_OUT.Locked=true
PD1-OSC_OUT.Mode=HSE-External-Clock-Source
PD1-OSC_OUT.Signal=RCC_OSC_OUT
PinOutPanel.RotationAngle=0
ProjectManager.AskForMigrate=true
ProjectManager.BackupPrevious=false
ProjectManager.CompilerOptimize=6
ProjectManager.ComputerToolchain=false
ProjectManager.CoupleFile=false
ProjectManager.CustomerFirmwarePackage=
ProjectManager.DefaultFWLocation=true
ProjectManager.DeletePrevious=true
ProjectManager.DeviceId=STM32F103RBTx
ProjectManager.FirmwarePackage=STM32CubeFW_F1 V1.8.5
ProjectManager.FreePins=false
ProjectManager.HalAssertFull=false
ProjectManager.HeapSize=0x200
ProjectManager.KeepUserCode=true
ProjectManager.LastFirmware=true
ProjectManager.LibraryCopy=0
ProjectManager.MainLocation=Core/Src
ProjectManager.NoMain=false
ProjectManager.PreviousToolchain=
ProjectManager.ProjectBuild=false
ProjectManager.ProjectFileName=Lab4.ioc
ProjectManager.ProjectName=Lab4
ProjectManager.ProjectStructure=
ProjectManager.RegisterCallBack=
ProjectManager.StackSize=0x400
ProjectManager.TargetToolchain=MDK-ARM V5.32
ProjectManager.ToolChainLocation=
ProjectManager.UAScriptAfterPath=

```



```

ProjectManager.UAScriptBeforePath=
ProjectManager.UnderRoot=false
ProjectManager.functionlistsort=1-
SystemClock_Config-RCC=false-HAL=false, 2-
MX_GPIO_Init-GPIO=false-HAL=true, 3-
MX_DMA_Init-DMA=false-HAL=true, 4-
MX_USART2_UART_Init-USART2=false-HAL=true, 5-
MX_USART1_UART_Init-USART1=false-HAL=true
RCC.ADCFreqValue=32000000
RCC.AHBFreq_Value=64000000
RCC.APB1CLKDivider=RCC_HCLK_DIV2
RCC.APB1Freq_Value=32000000
RCC.APB1TimFreq_Value=64000000
RCC.APB2Freq_Value=64000000
RCC.APB2TimFreq_Value=64000000
RCC.FCLKCortexFreq_Value=64000000
RCC.FamilyName=M
RCC.HCLKFreq_Value=64000000
RCC.IPPParameters=ADCFreqValue,AHBFreq_Value,A
PB1CLKDivider,APB1Freq_Value,APB1TimFreq_Valu
e,APB2Freq_Value,APB2TimFreq_Value,FCLKCortex
Freq_Value,FamilyName,HCLKFreq_Value,MCOFreq_
Value,PLLCLKFreq_Value,PLLMCOFreq_Value,PLLMU
L,RTCClockSelection,RTCFreq_Value,SYSCCLKFreq_
VALUE,SYSCCLKSource,TimSysFreq_Value,USBFreq_V
alue,VCOOutput2Freq_Value
RCC.MCOFreq_Value=64000000
RCC.PLLCLKFreq_Value=64000000
RCC.PLLMCOFreq_Value=32000000
RCC.PLLMUL=RCC_PLL_MUL16
RCC.RTCClockSelection=RCC_RTCCCLKSOURCE_LSE
RCC.RTCFreq_Value=32768
RCC.SYSCCLKFreq_VALUE=64000000
RCC.SYSCCLKSource=RCC_SYSCCLKSOURCE_PLLCLK
RCC.TimSysFreq_Value=64000000
RCC.USBFreq_Value=64000000
RCC.VCOOutput2Freq_Value=4000000
SH.GPXTI13.0=GPIO_EXTI13
SH.GPXTI13.ConfNb=1
USART1.BaudRate=9600
USART1.IPPParameters=VirtualMode,BaudRate
USART1.VirtualMode=VM_ASYNC
USART2.BaudRate=9600
USART2.IPPParameters=VirtualMode,BaudRate
USART2.VirtualMode=VM_ASYNC
VP_SYS_VS_Systick.Mode=SysTick
VP_SYS_VS_Systick.Signal=SYS_VS_Systick
board=NUCLEO-F103RB
boardIOC=true

```