

B-L47S5I-IOT01A ALL SENSORS DATA TRANSFER OVER WIFI

@Verbatim

* @file Documentation.pdf

* @author Sheraz Hassan

* @brief Description of the Es-WIFI Module in Local server mode transmitting data from all the integrated sensors including microphones from B-L4S5I-IOT01A module to the local server at specific rates

* STM Microelectronics reserves the rights to the Es-WIFI Module part of the software

* This software component is licensed by ST under Ultimate Liberty license SLA0044, the "License"; You may not use this file except in compliance with the License.

* You may obtain a copy of the License at:

<http://www.st.com/SLA0044>

@endverbatim

Application Description

This application shows how to make requests to a local server and transmit data to it using the Es-WiFi module based on STM32Cube HAL.

Step1-

Enter the local IP address to the main.c file along with the local port you wish to create on the server and enter the WiFi password you want to work on in the main.c file and rebuild the project.

Step2-

Connect the board ST-Link USB port to a PC USB port and upload the code on the IOT module. On the PC, start any Terminal application (i.e Termit) and configure It to the following

Terminal configuration:

- Data Length = 8 Bits
- One Stop Bit
- Parity = None
- Baud Rate = 115200 baud
- Flow control = None

Push the reset button, a virtual COM port will then appear in the Terminal. Start a WiFi Access Point on a smartphone or use a WiFi router. On virtual COM port enter the SSID and the type of encryption you are using. The WiFi configuration settings are stored in Flash memory. They will be retrieved from Flash memory the next time the board is reset. If you need to change the stored settings, during start the application proposes to change them by pressing the board's blue button ("User" button).

Note: In case of Open WiFi you don't need to enter password in Step1

Step3-

Now if a remote port is operational on the server and the server is connected on the same WiFi, you will start receiving the data once the module connects to the server.

Data transfer rate

- All the Sensors except microphones are designed to send data to the server @ 1k/min.
- Microphone is operating @8.1kHz.

Received data

Received data will be in the following form

- [xx-xx-xx xx:xx:xx.xx] == Timestamp (Formatting may vary as timestamp is created by the server)
- (xxxxx) or (-xxxxx) == Microphone readings [x represents a digit of the reading]
- |xxxx.x| == Temperature (Celsius) [x represents a digit of the reading]
- !xxxx.x! == Humidity (relative %) [x represents a digit of the reading]
- \$xxxx.x\$ == Pressure (mbar) [x represents a digit of the reading]
- *xxxxx* == Distance (mm) [x represents a digit of the reading]
- ^-----.-x-----.-x----- ^ == acceleration (xaxis,yaxis,zaxis)(g) [- is a digit of the reading]
- @^-----.-x-----.-x----- @ == Gyroscope (xaxis,yaxis,zaxis)(Degrees)
- #^-----.-x-----.-x----- # == Magnetometer (xaxis,yaxis,zaxis)(Degrees)

Directory contents

- /Inc/main.h	Header for main.c module
- /Inc/stm32l4xx_hal_conf.h	HAL configuration file
- /Inc/stm32l4xx_it.h	STM32 interrupt handlers header file
- /Src/main.c	Main Program
- /Src/flash_l4.c	Flash management (for storing configuration)
- /Src/system_stm32l4xx.c	STM32L4xx system clock configuration file
- /Src/stm32l4xx_it.c	STM32 interrupt handlers
- /Common/Src/es_wifi.c	Implementation of the ES_WIFI_XXX() API.
- /Common/Src/es_wifi_io.c	Implementation of the ES_WIFI_IO_XXX() API.

- /Common/Src/wifi.c Implementation of the WIFI_XXX() API.
- /Common/Inc/es_wifi.h Header for the functions and defines used by the es_wifi.c
- /Common/Inc/es_wifi_io.h Header for the functions and defines used by the es_wifi_io.c
- /Common/Inc/wifi.h Header for the functions and defines used by the wifi.c

Hardware environment

This application has been tested on the following environment:

- B-L4S5I-IOT01A board