Cloud-JAM L4 board

Data sheet

27/10/2017

Summary

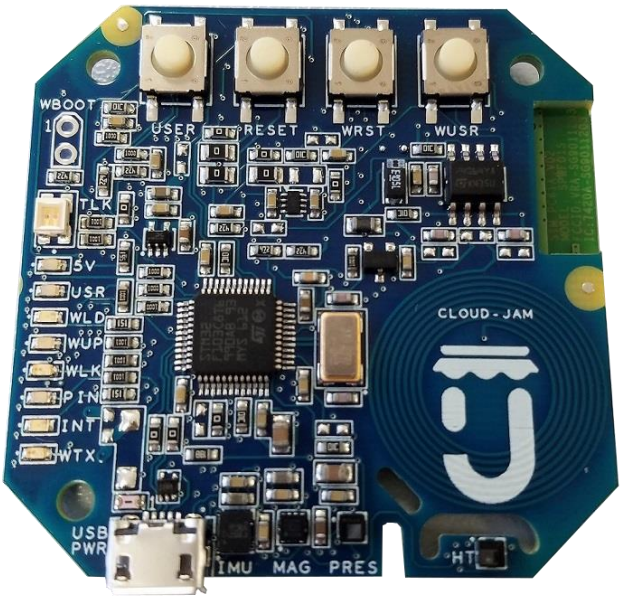
.1 REVISION HISTORY 1
.2 CLOUD-JAM L4 BOARD IMAGES 2
.3 INTRODUCTION 3
.4 DESCRIPTION 3
.5 BLOCK DIAGRAM 4
.6 HARDWARE & COMPONENTS DETAIL 5
.6.1 Embedded debugger 5
.6.2 CPU 5
.6.3 Sensors 6
.6.4 NFC memory 10
.6.5 Wi-Fi module SPWF01SA 10
.6.6 LEDs 11
.6.7 Push buttons 11
.6.8 USB connector 12
.6.9 Expansions connector 13
.7 DEVELOPMENT & DEBUG WITH CLOUD-JAM L4 13
.7.1 System requirements 13
.7.2 IDEs supporting STM32 core inside Cloud-JAM L4 14
.7.3 ST-LINK/V2-1 installation 14
.7.4 Firmware package 14
.7.5 Executing and debugging firmware using software toolchains 15
.8 GO TO PRODUCTION WITH CLOUD-JAM L4 16
.8.1 ST-LINK UTILITY PC REQUIREMENTS 17
.8.2 Installing the STM32 ST-LINK utility 17
.8.3 Uninstalling the STM32 ST-LINK utility 17
.8.4 Cloud-JAM L4 programming using ST-LINK UTILITY 17
.9 TYPICAL APPLICATION 18
.9.1 FP-CLD-AZURE1 software description 19
.9.2 Use NFC to configure Wi-Fi Access Point parameters 24
.10 ELECTRICAL SPECIFICATIONS 25
.11 MECHANICAL SPECIFICATIONS 26
.12 HOUSING PRESCRIPTION 26
.13 PACKAGING 27
.14 UE DECLARATION OF CONFORMITY 28
.15 OPERATING ENVIRONMENT 29
.16 DICLAIMERS 29

.1 REVISION HISTORY

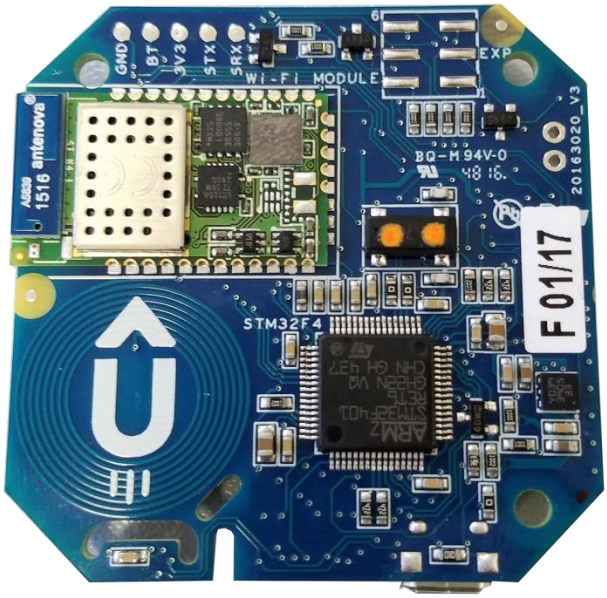
Date	Revision	Revision history
09/06/2017	1	Draft version
29/06/2017	2	Expansion connetor pinout is different, there is no i2c or uart in Cloud-JAM L4
27/10/2017	3	Added EU DoC

.2 CLOUD-JAM L4 BOARD IMAGES

TOP VIEW



BOTTOM VIEW



.3 INTRODUCTION

This data sheet provides the description of the Cloud-JAM L4 board, a member of the JAM family boards.
As RushUp product accelerator, Cloud-JAM L4 can be used from makers, developers, high mix low volume products and from all who want to benefit from an off the shelf industrialized board.
In particular Cloud-JAM L4 is the product accelerator of the STM32 ODE system.
For details about RushUp and STM32 ODE, please visit:

www.rushup.tech

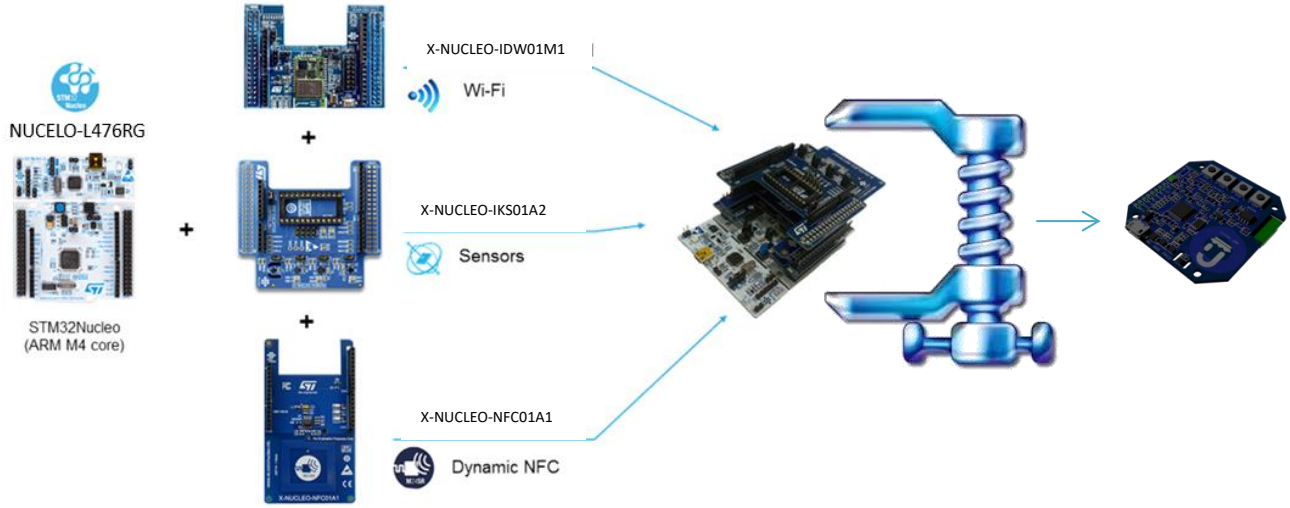
http://www.st.com/content/st_com/en/products/ecosystems/stm32-open-development-environment.html

.4 DESCRIPTION

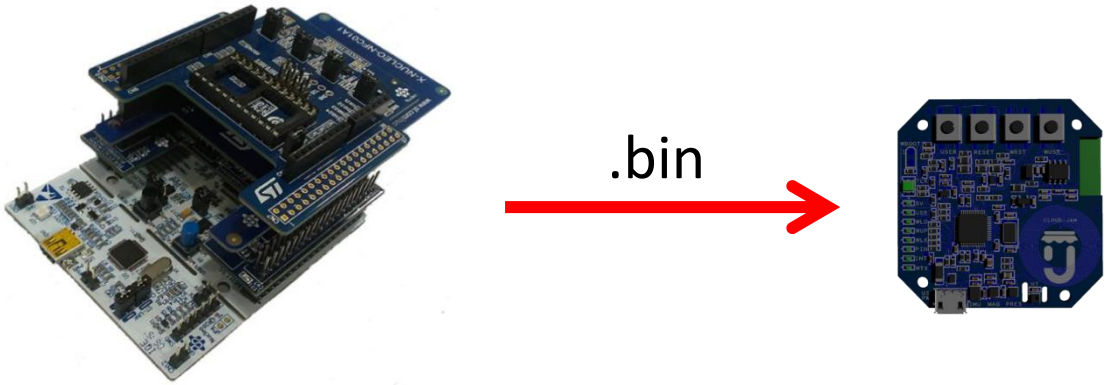
The Cloud-JAM L4 board is the industrialized version of the FP-CLD-AZURE1 STM32 ODE functional pack. It can connect IoT node based on STM32 Nucleo to Microsoft Azure cloud, transmits sensor data and receive commands from Cloud application.
Motion & environmental sensors board connected to the cloud through Wi-Fi network using SSID, Password and web authentication stored in the dynamic NFC.
Inside Cloud-JAM L4 you can find the following STM32-Nucleo elements:

- NUCLEO-L476RG
- X-NUCLEO-IDW01M1
- X-NUCLEO-IKS01A2
- X-NUCLEO-NFC01A1

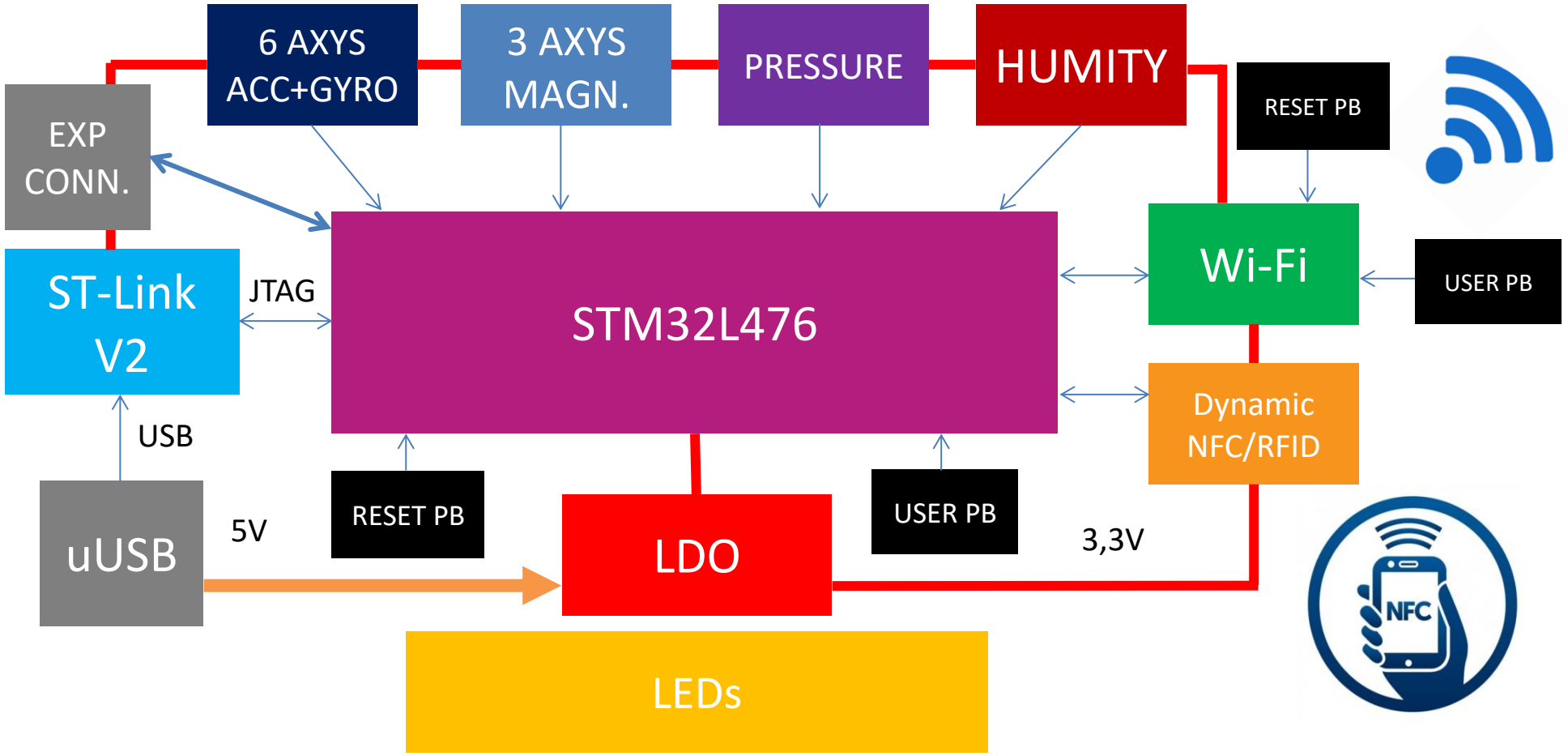
The connection and the components are the same as all the above boards connected together in a single functional pack as indicated by the next picture.



Cloud-JAM L4 board embeds also the debugger so it can be used also for development and debug.
Since the board is exactly the same as the functional pack (from electrical connections point of view), one of the most importat features is that the firmware is plug & play and compatible between the two development systems. User can develop and debug on Nucleo functional pack and then download the .bin file into the Cloud-JAM L4 board.



.5 BLOCK DIAGRAM



.6 HARDWARE & COMPONENTS DETAIL

.6.1 Embedded debugger

Cloud-JAM L4 incorporates the ST-LINK V2 debugger and then is capable not only to download the firmware but also to debug it with the main development tools.
The ST-LINK/V2 is an in-circuit debugger and programmer for the STM8 and STM32 microcontroller families. The single wire interface module (SWIM) and JTAG/serial wire debugging (SWD) interfaces are used to communicate with any STM8 or STM32 microcontroller located on an application board.
STM32 applications use the USB full-speed interface to communicate with Atollic®, IAR™, Keil® or TASKING integrated development environments.
Direct firmware update feature supported (DFU) and Status LED which blinks during communication with the PC.

.6.2 CPU

The STM32L476 device is an ultra-low-power microcontroller based on the high-performance ARM® Cortex®-M4 32-bit RISC core operating at a frequency of up to 80 MHz. The Cortex-M4 core features a Floating point unit (FPU) single precision which supports all ARM single-precision data-processing instructions and data types. It also implements a full set of DSP instructions and a memory protection unit (MPU) which enhances application security.

The STM32L476 device embed high-speed memories (Flash memory 1 Mbyte, 128 Kbyte of SRAM), a flexible external memory controller (FSMC) for static memories (for devices with packages of 100 pins and more), a Quad SPI flash memories interface (available on all packages) and an extensive range of enhanced I/Os and peripherals connected to two APB buses, two AHB buses and a 32-bit multi-AHB bus matrix.

The STM32L476 device embed several protection mechanisms for embedded Flash memory and SRAM: readout protection, write protection, proprietary code readout protection and Firewall.

The devices offer up to three fast 12-bit ADCs (5 Msps), two comparators, two operational amplifiers, two DAC channels, an internal voltage reference buffer, a low-power RTC, two general-purpose 32-bit timer, two 16-bit PWM timers dedicated to motor control, seven general-purpose 16-bit timers, and two 16-bit low-power timers. The devices support four digital filters for external sigma delta modulators (DFSDM).

In addition, up to 24 capacitive sensing channels are available. The devices also embed an integrated LCD driver 8x40 or 4x44, with internal step-up converter.

- Ultra-low-power with FlexPowerControl
  - 1.71 V to 3.6 V power supply
  - -40 °C to 85/105/125 °C temperature range
  - 300 nA in VBAT mode: supply for RTC and 32x32-bit backup registers
  - 30 nA Shutdown mode (5 wakeup pins)
  - 120 nA Standby mode (5 wakeup pins)
  - 420 nA Standby mode with RTC
  - µA Stop 2 mode, 1.4 µA Stop 2 with RTC
  - 100 µA/MHz run mode
  - Batch acquisition mode (BAM)
  - 4 µs wakeup from Stop mode
  - Brown out reset (BOR) in all modes except shutdown
  - Interconnect matrix
- Core: ARM® 32-bit Cortex®-M4 CPU with FPU, Adaptive real-time accelerator (ART Accelerator™) allowing 0-wait-state execution from Flash memory, frequency up to 80 MHz, MPU, 100DMIPS/1.25DMIPS/MHz (Dhrystone 2.1), and DSP instructions Clock Sources
  - 4 to 48 MHz crystal oscillator
  - 32 kHz crystal oscillator for RTC (LSE)
  - Internal 16 MHz factory-trimmed RC (±1%)
  - Internal low-power 32 kHz RC (±5%)
  - Internal multispeed 100 kHz to 48 MHz oscillator, auto-trimmed by LSE (better than ±0.25 % accuracy)

  - 3 PLLs for system clock, USB, audio, ADC
- RTC with HW calendar, alarms and calibration
- LCD 8 × 40 or 4 × 44 with step-up converter
- Up to 24 capacitive sensing channels: support touchkey, linear and rotary touch sensors
- 16x timers: 2 x 16-bit advanced motor-control, 2 x 32-bit and 5 x 16-bit general purpose, 2x 16-bit basic, 2x low-power 16-bit timers (available in Stop mode), 2x watchdogs, SysTick timer
- Up to 114 fast I/Os, most 5 V-tolerant, up to 14 I/Os with independent supply down to 1.08 V
- Memories
  - Up to 1 MB Flash, 2 banks read-while-write, proprietary code readout protection
  - Up to 128 KB of SRAM including 32 KB with hardware parity check
  - External memory interface for static memories supporting SRAM, PSRAM, NOR and NAND memories
- Quad SPI memory interface
- 4x digital filters for sigma delta modulator
- Rich analog peripherals (independent supply)
- 3× 12-bit ADC 5 Msps, up to 16-bit with hardware oversampling, 200 µA/Msps
- 2x 12-bit DAC, low-power sample and hold
- 2x operational amplifiers with built-in PGA
- 2x ultra-low-power comparators
- 18x communication interfaces
- USB OTG 2.0 full-speed, LPM and BCD
- 2x SAIs (serial audio interface)
- 3x I2C FM+(1 Mbit/s), SMBus/PMBus
- 6x USARTs (ISO 7816, LIN, IrDA, modem)
- 3x SPIs (4x SPIs with the Quad SPI)
- CAN (2.0B Active) and SDMMC interface
- SWPMI single wire protocol master I/F
- 14-channel DMA controller
- True random number generator
- CRC calculation unit, 96-bit unique ID

.6.3 Sensors

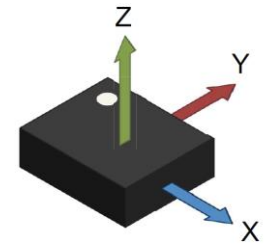
.6.3.1 iNEMO inertial module LSM6DSL

The LSM6DSL is a system-in-package featuring a 3D digital accelerometer and a 3D digital gyroscope performing at 0.65 mA in high-performance mode and enabling always-on low-power features for an optimal motion experience for the consumer. .
The LSM6DSL supports main OS requirements, offering real, virtual and batch sensors with 4 kbyte for dynamic data batching.
ST's family of MEMS sensor modules leverages the robust and mature manufacturing processes already used for the production of micromachined accelerometers and gyroscopes.
The various sensing elements are manufactured using specialized micromachining processes, while the IC interfaces are developed using CMOS technology that allows the design of a dedicated circuit which is trimmed to better match the characteristics of the sensing element.
The LSM6DSL has a full-scale acceleration range of ±2/±4/±8/±16 g and an angular rate range of ±125/±245/±500/±1000/±2000 dps. High robustness to mechanical shock makes the LSM6DSL the preferred choice of system designers for the creation and manufacturing of reliable products.
The LSM6DSL is available in a plastic land grid array (LGA) package

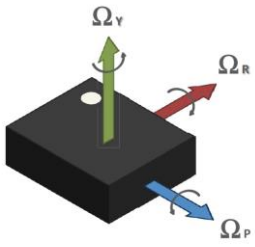
Features

- Power consumption: 0.4 mA in combo normal mode and 0.65 mA in combo high-performance mode
- "Always-on" experience with low power consumption for both accelerometer and gyroscope
- Smart FIFO up to 4 kbyte based on features set
- Compliant with Android K, L, and M
- Hard, soft ironing for external magnetic sensor corrections
- ±2/±4/±8/±16 g full scale

- ±125/±245/±500/±1000/±2000 dps full scale
- Analog supply voltage: 1.71 V to 3.6 V
- Independent IOs supply (1.62 V)
- Compact footprint, 2.5 mm x 3 mm x 0.83 mm
- SPI & I2C serial interface with main processor data synchronization feature
- Pedometer, step detector and step counter
- Significant motion and tilt function
- Standard interrupts: free-fall, wakeup, 6D/4D orientation, click and double-click
- Embedded temperature sensor
- ECOPACK®, RoHS and “Green” compliant



Direction of detectable acceleration (top view)



Direction of detectable angular rate (top view)

For any specific information and for the firmware commands and procedures, please refer to the data sheet of the components.

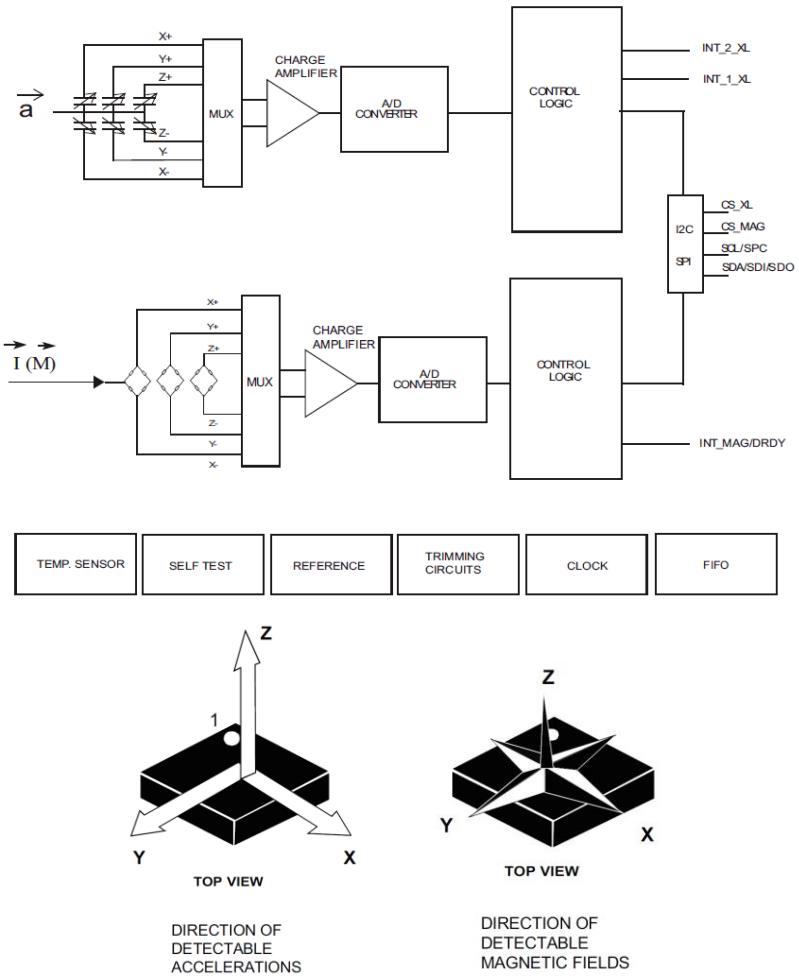
.6.3.2 High-performance eCompass module LSM303AGR

The LSM303AGR is an ultra-low-power highperformance system-in-package featuring a 3D digital linear acceleration sensor and a 3D digital magnetic sensor.
The LSM303AGR has linear acceleration full scales of ±2g/±4g/±8g/±16g and a magnetic field dynamic range of ±50 gauss.
The LSM303AGR includes an I2C serial bus interface that supports standard, fast mode, fast mode plus, and high-speed (100 kHz, 400 kHz, 1 MHz, and 3.4 MHz) and an SPI serial standard interface.
The system can be configured to generate an interrupt signal for free-fall, motion detection and magnetic field detection.
The magnetic and accelerometer blocks can be enabled or put into power-down mode separately. The LSM303AGR is available in a plastic land grid array package (LGA) and is guaranteed to operate over an extended temperature range from -40 °C to +85 °C.

Features

- 3 magnetic field channels and 3 acceleration channels
- ±50 gauss magnetic dynamic range
- ±2/±4/±8/±16 g selectable acceleration full scales
- 16-bit data output
- SPI / I2C serial interfaces
- Analog supply voltage 1.71 V to 3.6 V
- Selectable power mode/resolution for accelerometer and magnetometer
- Single measurement mode for magnetometer
- Programmable interrupt generators for freefall, motion detection and magnetic field detection
- Embedded self-test
- Embedded temperature sensor
- Embedded FIFO
- ECOPACK®, RoHS and “Green” compliant

Block diagram



For any specific information and for the firmware commands and procedures, please refer to the data sheet of the components.

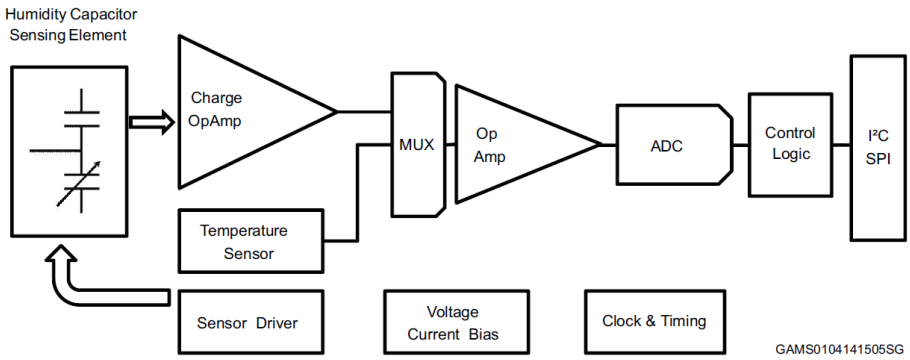
.6.3.3 Humidity and temperature

The HTS221 is an ultra-compact sensor for relative humidity and temperature. It includes a sensing element and a mixed signal ASIC to provide the measurement information through digital serial interfaces.
The sensing element consists of a polymer dielectric planar capacitor structure capable of detecting relative humidity variations and is manufactured using a dedicated ST process.
The HTS221 is available in a small top-holed cap land grid array (HLGA) package guaranteed to operate over a temperature range from -40 °C to +120 °C.

Features

- 0 to 100% relative humidity range
- Supply voltage: 1.7 to 3.6 V
- Low power consumption: 2 µA @ 1 Hz ODR
- Selectable ODR from 1 Hz to 12.5 Hz
- High rH sensitivity: 0.004% rH/LSB
- Humidity accuracy: ± 3.5% rH, 20 to +80% rH
- Temperature accuracy: ± 0.5 °C,15 to +40 °C
- Embedded 16-bit ADC
- 16-bit humidity and temperature output data
- SPI and I²C interfaces
- Factory calibrated
- Tiny 2 x 2 x 0.9 mm package
- ECOPACK® compliant

Block diagram



For any specific information and for the firmware commands and procedures, please refer to the data sheet of the components.

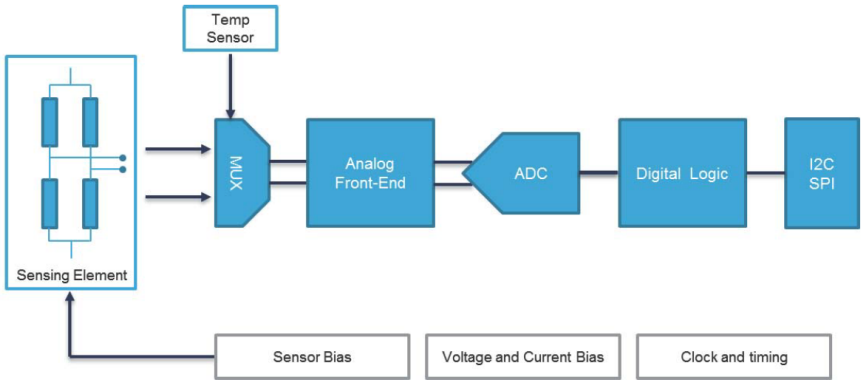
.6.3.4 Barometer

The LPS22HB is an ultra-compact piezo resistive absolute pressure sensor which functions as a digital output barometer. The device comprises a sensing element and an IC interface which communicates through I2C or SPI from the sensing element to the application.
The sensing element, which detects absolute pressure, consists of a suspended membrane manufactured using a dedicated process developed by ST.
The LPS22HB is available in a full-mold, holed LGA package (HLGA). It is guaranteed to operate over a temperature range extending from -40 °C to +85 °C. The package is holed to allow external pressure to reach the sensing element.

Features

- 260 to 1260 hPa absolute pressure range
- Current consumption down to 3 µA
- High overpressure capability: 20x full-scale
- Embedded temperature compensation
- 24-bit pressure data output • 16-bit temperature data output
- ODR from 1 Hz to 75 Hz
- SPI and I²C interfaces
- Embedded FIFO
- Interrupt functions: Data Ready, FIFO flags, pressure thresholds
- Supply voltage: 1.7 to 3.6 V
- High shock survivability: 22,000 g
- Small and thin package • ECOPACK® lead-free compliant

Block diagram



For any specific information and for the firmware commands and procedures, please refer to the data sheet of the components.

.6.4 NFC memory

The M24SR64-Y device is a dynamic NFC/RFID tag that can be accessed either from the I2C or the RF interface. The RF and I2C host can read or write to the same memory, that is why only one host can communicate at a time with the M24SR64-Y. The management of the interface selection is controlled by the M24SR64-Y device itself. The RF interface is based on the ISO/IEC 14443 Type A standard. The M24SR64-Y is compatible with the NFC Forum Type 4 Tag specifications and supports all corresponding commands. The I2C interface uses a two-wire serial interface consisting of a bidirectional data line and a clock line. The devices carry a built-in 4-bit device type identifier code in accordance with the I²C bus definition. The device behaves as a slave in the I2C protocol.

I2C mode

M24SR64-Y is powered by VCC. The I2C interface is connected to the M24SR64-Y. The I2C host can communicate with the M24SR64-Y device. 1.1.2

Tag mode

The M24SR64-Y is supplied by the RF field and can communicate with an RF host (RFID reader or an NFC phone). The User memory can only be accessed by the RF commands.

Dual interface mode

Both interfaces, RF and I2C, are connected to the M24SR64-Y and both RF or I2C host can communicate with the M24SR64-Y device. The power supply and the access management are carried out by the M24SR64-Y itself. For further details, please refer to the token mechanism described inside the data sheet of the component.

- Contactless interface
- NFC Forum Type 4 Tag
- ISO/IEC 14443 Type A
- 106 Kbps data rate
- Internal tuning capacitance: 25 pF Memory
- 8-Kbyte (64-kbit) EEPROM
- Support of NDEF data structure
- Data retention: 200 years
- Write cycle endurance:
  - 1 million Write cycles at 25 °C
  - 600k Write cycles at 85 °C
  - 500k Write cycles at 105 °C
- Read up to 246 bytes in a single command
- Write up to 246 bytes in a single command
- 7 bytes unique identifier (UID)
- 128 bits passwords protection

For any specific information and for the firmware commands and procedures, please refer to the data sheet of the components.

.6.5 Wi-Fi module SPWF01SA

The SPWF01SA intelligent Wi-Fi module represent a plug-and-play and standalone 802.11 b/g/n solution for easy integration of wireless Internet connectivity features into existing or new products.
Configured around a single-chip 802.11 transceiver with integrated PA and comprehensive power management subsystem, and an STM32 microcontroller with an extensive GPIO suite, the modules also incorporate timing clocks and voltage regulators.
The SPWF01SA.11 orderable parts integrate 1.5 MB of Flash and is available with an embedded micro 2.45 GHz highly-efficient ISM band antenna.
With low power consumption and ultra-compact (2.7 x 1.5 cm) footprint, the module is ideal for fixed and mobile wireless applications, as well as challenging battery-operated applications.
The SPWF01SA parts is released with an integrated full featured TCP/IP protocol stack with added web server and additional application service capabilities, such as REST API for accessing files on servers in the cloud and support for dynamic web pages with CGI/SSI functions to easily interact with the module and the host processor over the air.
For secure end-to-end communication with the cloud, an SSL/TLS stack is embedded in every module with no licensing charge. See application note AN4683 for details.
The SW package also includes an AT command layer interface for user-friendly access to the stack functionalities via the UART serial port. For details, see user manual UM1695.
It is always possible to upgrade the module firmware via UART and Over The Air (FOTA). Out of the 1 MB extended Flash available on the SPWF01Sx.1y module, 512 KB of the Flash is dedicated to FW upgrade and the other 512 KB is dedicated to host proprietary files, organized in a file system image, accessible

through the integrated web server.. ST recommends that users regularly check for documentation and the current FW version available at www.st.com/wifimodules.
This section provides some information regarding the SPWF01SA Serial-to-Wi-Fi b/g/n intelligent module (order code: SPWF01SA.11). More detailed information can be found at the www.st.com web site.

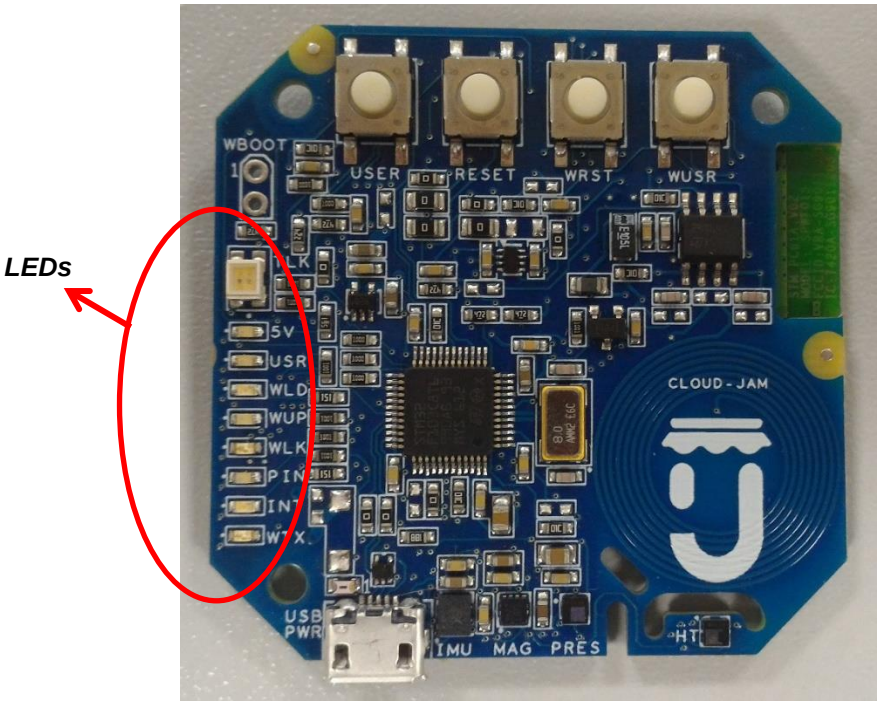
The main features of the SPWF01SA module are:

- 2.4 GHz IEEE 802.11 b/g/n transceiver
- STM32 ARM Cortex-M3, with 64 KB RAM and 512KB Flash memory
  - 1 MByte extended FLASH for FW update
  - Over the air (FOTA)
- Integrated TCP/IP protocol stack
  - 8 Simultaneous TCP or UDP clients and 1 Socket server
  - 1 TLS/SSL Socket client supporting up to TLS 1.2, including common encryption algorithms: AES (128,256), hash (MD5, SHA-1, SHA-256) and public key algorithms (RSA, ECC).
  - Web server supporting dynamic web pages
  - RESTful API to get & post web content
- WEP/WPA/WPA2 personal security
- System modes: station, IBSS, and miniAP
- miniAP easily provisioned (SSID, PWD)
- Industrial temperature range: -40 °C to 85 °C
- SPWF01SA is a fully FCC/IC/CE certified module.

.6.6 LEDs

Cloud-JAM L4 integrates all the LEDs presents in the Nucleo functional pack, in particular they are:

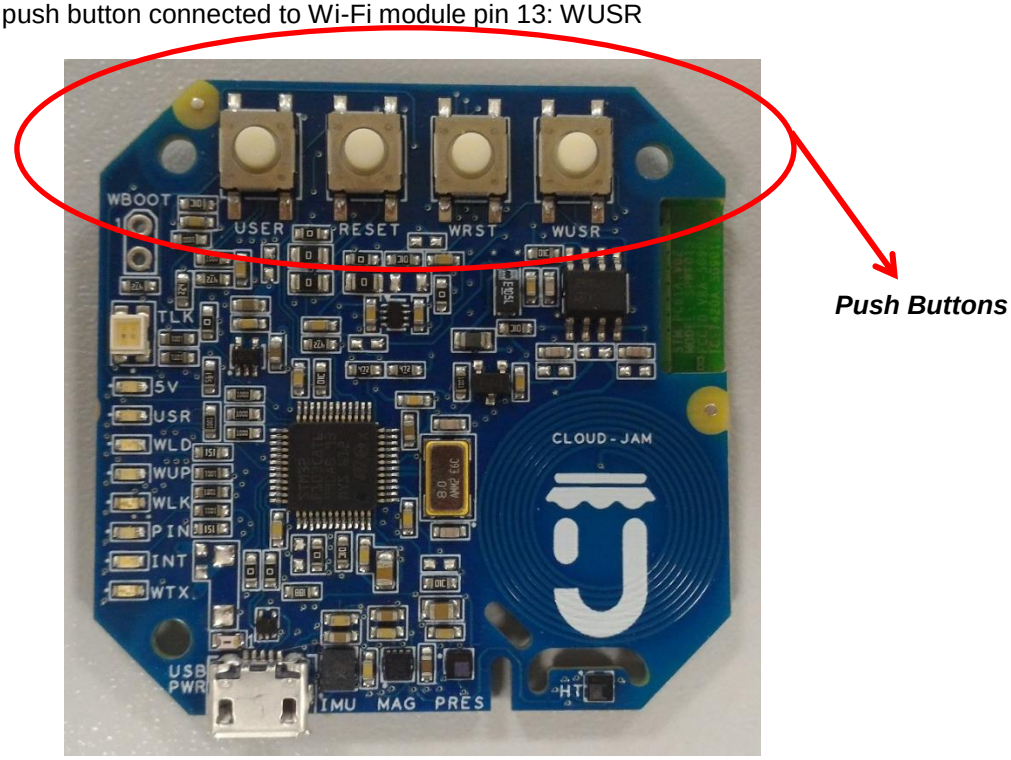
- ST-LINK V2 bicolor LED: TLK LED
- Power supply present: 5V LED
- User free connected to MCU pin PA5: USR LED
- Wi-Fi blink connected to pin 5 of the module: WLD LED
- Wi-Fi power up connected to pin 14 of the module: WUP LED
- Wi-Fi link connected to pin 15 of the module: WLK LED
- iNEMO interrupt 2 connected to pin 9 of LSM6DSL: PIN LED
- iNEMO interrupt 1 connected to pin 4 of LSM6DSL: INT LED
- Wi-Fi TXD1 connected to pin 6 of the module: WTX LED



.6.7 Push buttons

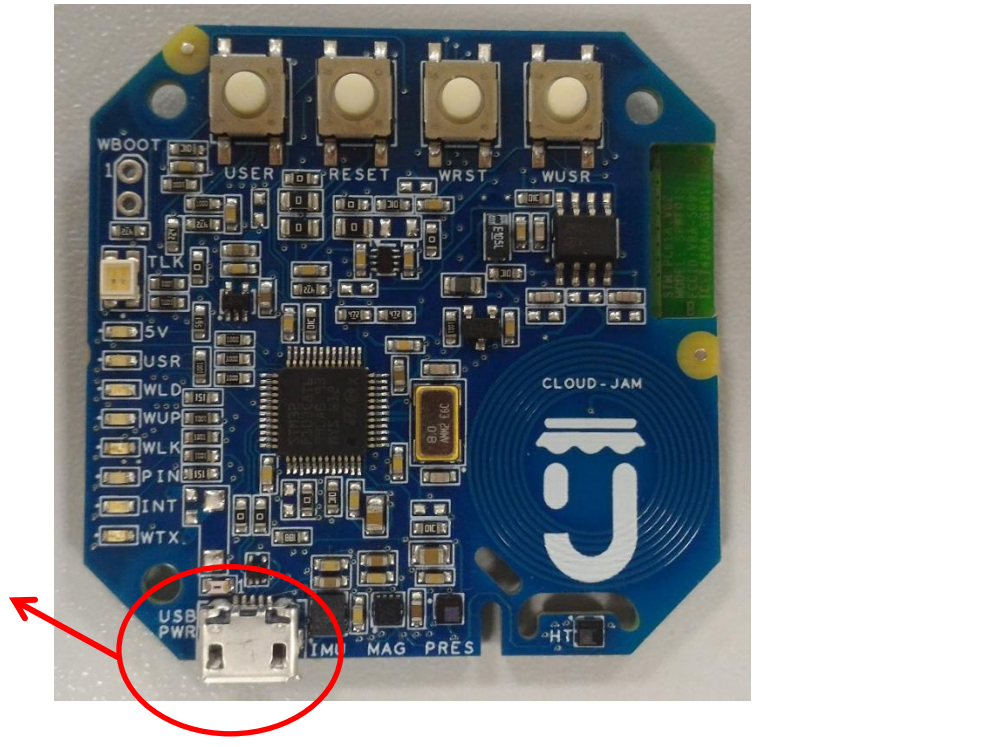
Cloud-JAM L4 integrates alla the push buttons presents in the Nucleo functional pack, in paticular they are:

- PB1 user push button connected to MCU pin PC13: USER
- PB2 reset push button connected to MCU pin NRST: RESET
- PB3 reset push button connected to Wi-Fi module pin nRESET: WRST
- PB4 user push button connected to Wi-Fi module pin 13: WUSR



.6.8 USB connector

The USB connector act as power supply input porta at 5V dc level and also as debugger and programmer as refered to chapter 5.1.
The connector is USB female 90° orientation micro B type and the part number mounted on the board is the FCI 10118194-0001LF.



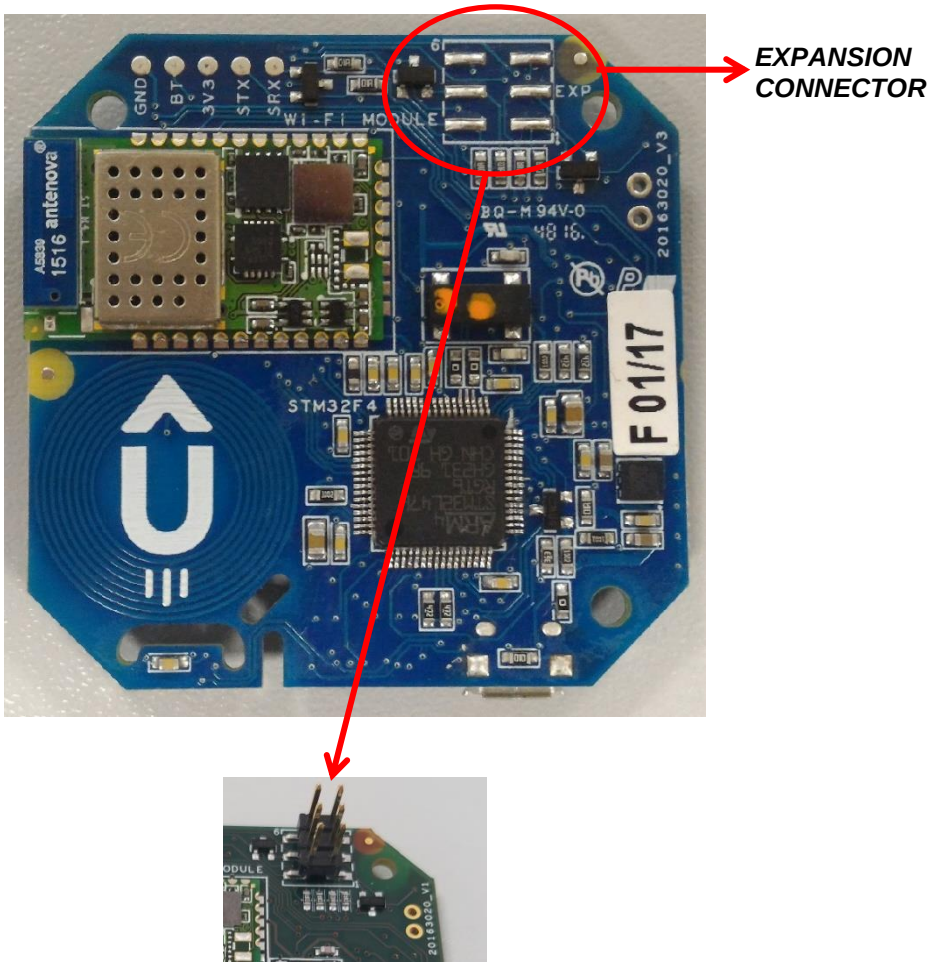
.6.9 Expansions connector

Even if Cloud-JAM L4 is mainly intended as industrialized version of FP-CLD-BLUEMIX1 Nucleo functional pack, there is also an expansion connector in which there are power supply and four general purpose I/O.

Connector is vertical type 2.54mm pitch double row Amphenol part number 95278-101A06LF

Here in after the connection table.

PIN NUMBER	NAME	DESCRIPTION
1	GPIO/PA8	PA8 pin as GPIO
2	GPIO/PC7	PC7 pin as GPIO
3	GPIO/PC9	PC9 pin as GPIO
4	GPIO/PC6	PC6 pin as GPIO
5	+3V3	3,3V power supply from Cloud-JAM L4. This is not intended as external power supply but only as voltage signal for polarization or logic circuits. Max current available for sourcing 25mA.
6	GND	GND reference voltage of the Cloud-JAM L4



.7 DEVELOPMENT & DEBUG WITH CLOUD-JAM L4

.7.1 System requirements

In case Cloud-JAM L4 is used to develop and debug applications usign ST-LINK V2 embedded debugger, before starting the user should:

1. Install the preferred Integrated Development Environment (IDE).
2. ST-LINK/V2-1 driver will be installed automatically. In case of problem, the user can proceed with manual installation of the driver, from toolchains install directory (further details are available in Section 3).
3. Download the STM32 Nucleo firmware from the www.st.com/stm32nucleo webpage.
4. Establish the connection with the STM32 Nucleo board, by connecting the board to the USB port of the PC.

The above steps will be detailed in the coming sections

.7.2 IDEs supporting STM32 core inside Cloud-JAM L4

STMicroelectronics STM32 32-bit ARM® Cortex® -M core-based microcontrollers are supported by a complete range of software tools. It encompasses traditional integrated development environment IDEs with C/C++ compilers and debuggers from major third-parties (free versions up to 64 Kbytes of code, depending on third-party), completed with innovative tools from STMicroelectronics.
Toolchains supporting all Nucleo boards are:

- EWARM v7.10.3 or later(a)
  - 30-day evaluation edition
  - 32-Kbyte Limited QuickStart edition (16-Kbyte limitation for Cortex M0)
- MDK-ARM v5.17 or later(a)(b)
  - MDK-Lite (32-Kbyte code size limitation)
- TrueSTUDIO Lite v5 or later(b)
  - No limitation
- SW4STM32 v1.5 and later(a)
  - No limitation

Information on the toolchain version supporting the STM32 device is available on toolchain release note at third-party web site.

.7.3 ST-LINK/V2-1 installation

Exactly as per all the STM32 Nucleo boards, Cloud-JAM L4 include an ST-LINK/V2-1 embedded debug tool interface. This interface needs a dedicated USB driver to be installed. This driver is available at the www.st.com website and supported within the following software toolchains:

- IAR Embedded Workbench for ARM (EWARM). The toolchain is installed by default in the C:\Program Files\IAR Systems\Embedded Workbench x.x directory on the PC local hard disk. After installing EWARM, install the ST-LINK/V2-1 driver by running the ST-Link_V2_USB.exe from [IAR_INSTALL_DIRECTORY]\Embedded Workbench x.x\arm\drivers\ST-Link \ST-Link_V2_USBdriver.exe.
- Keil Microcontroller Development Kit (MDK-ARM) toolchain. The toolchain is installed by default in the C:\Keil directory on the PC local hard disk; the installer creates a start menu µVision5 shortcut. When connecting the ST-LINK/V2-1 tool, the PC detects new hardware and requires to install the ST-LINK_V2_USB driver. The “Found New Hardware wizard” appears and guides the user through the steps needed to install the driver from the recommended location.
- Atollic TrueSTUDIO STM32. The toolchain is installed by default in the C:\Program Files\Atollic directory on the PC local hard disk. The ST-Link_V2_USB.exe file is automatically installed when installing the software toolchain.
- AC6 System Workbench for STM32 (SW4STM32). The toolchain is installed by default in the C:\Program Files\AC6 directory on the PC local hard disk. The ST-Link_V2_USB.exe file is automatically executed when installing the software toolchain.

Complementary information on the firmware package content and the STM32 Nucleo requirements is available on the Getting started with the STM32 Nucleo board firmware package UM1726 User manual at the www.st.com website.
Note: The embedded ST-LINK/V2-1 only supports SWD interface for STM32 core.

.7.4 Firmware package

The STM32 Nucleo examples, applications and demonstration are provided in one single .zip file. The extraction of the .zip file generates one folder, STM32 Nucleo_FW_VX.Y.Z, which contains the following subfolders:

- Template project is a pre-configured project with empty main function to be customized by the user. This is helpful to start creating an application based on the peripherals drivers.
- Example project includes toolchain projects for each peripheral example ready to be run. • Applications includes set of applications ready to be run.
- Demonstration includes the demonstration firmware ready to be run

.7.5 Executing and debugging firmware using software toolchains

The steps below can be applied to an already existing example, demonstration or template project available from STM32_Nucleo_FW_VX.Y.Z firmware at the www.st.com website. First of all, the user must read the firmware/readme.txt file, which contains the firmware description and hardware/software requirements.

.7.5.1 EWARM toolchain

The following procedure explains how to compile, link and execute an existing EWARM project.

1. Open IAR Embedded Workbench for ARM (EWARM).
2. In the File menu, select Open and click Workspace to display the Open Workspace dialog box. Browse to select either an example or demonstration or template workspace file and click Open to launch it in the Project window.
3. In the Project menu, select Rebuild All to compile the project.
4. If the project is successfully compiled, no errors and warning appears.

   To change the project settings (Include and preprocessor defines), go through the following project options:

   - For Include directories: ***Project>Options…>C/C++ compiler>***
   - For pre-processor defines: ***Project>Options…C/C++ compiler>pre-processor>***
5. In IAR Embedded Workbench IDE, from the Project menu, select Download and Debug or, alternatively, click the Download and Debug button in the toolbar, to program the Flash memory and start debugging.
6. The debugger in IAR Embedded Workbench can be used to debug source code at C and assembly levels, set breakpoints, monitor individual variables and watch events during the code execution. To run the application from the Debug menu, select Go. Alternatively, click the Go button in the toolbar to run the application.

.7.5.2 MDK-ARM toolchain

1. Open Keil MDK-ARM Microcontroller Development Kit. Figure 11 shows the basic names of the "Keil µvision5" windows, to which this document refers.
2. In the Project menu, select Open Project... Browse to select either an example or a demonstration or a template project file and click Open to launch it in the Project window.
3. In the Project menu, select Rebuild All target files to compile the project.
4. If the project is successfully compiled, the following window is displayed

   If user needs to change his project settings (Include and preprocessor defines), he must go through the project options:

   - For Include directories: ***Project>Options for Target > C/C++ > Include Paths***
   - For pre-processor defines: ***Project>Options for Target > C/C++ > Preprocessor symbols > Define***
5. In MDK-ARM IDE, from the Debug menu, select Start/Stop Debug Session or, alternatively, click the Start/Stop Debug Session button in the toolbar, to program the Flash memory and start debugging.
6. The debugger in the MDK-ARM can be used to debug source code at C and assembly levels, set breakpoints, monitor individual variables and watch events during the code execution.
   To run the application from the Debug menu, select Run. Alternatively, click the Run button in the toolbar to run the application.

.7.5.3 TrueSTUDIO toolchain

1. Open Atollic TrueSTUDIO for ARM product. The program launches and prompts for the workspace location.
2. Browse to select a TrueSTUDIO workspace of either an example or a demonstration or a template workspace file and click **OK** to load it.

3. To load an existing project in the selected workspace, select **Import** from the **File** menu to display the **Import** dialog box.
4. In the Import window, open General, select Existing Projects into Workspace and click Next.
5. Click Select root directory and browse to the TrueSTUDIO workspace folder.
6. In the Projects panel, select the project and click Finish.
7. In the Project Explorer, select the project, open the Project menu, and click Build Project.
8. If the project is successfully compiled, the following messages will be displayed on the console window

   If user needs to change the project settings (Include directories and preprocessor defines), he must go through Project>Properties and select C/C++ Build>Settings from the left panel:

   - For Include directories: C Compiler>Directories>Include path
   - For pre-processor defines: C Compiler>Symbols> Defined symbols

9. To debug and run the application, select the project in the Project Explorer and press F11 to start a debug session.

The debugger in the Atollic TrueSTUDIO can be used to debug source code at C and assembly levels, set breakpoints, monitor individual variables and watch events during the code execution. To run the application, from the Run menu, select Resume, or alternatively click the Resume button in the toolbar.

.7.5.4 SW4STM32 toolchain

1. Open the AC6 SW4STM32 for STM32. The program launches and prompts for the workspace location.
2. Browse to select a SW4STM32 workspace of either an example or a demonstration or a template workspace file and click OK to load it.
3. To load an existing project in the selected workspace, select Import from the File menu to display the Import dialog box.
4. In the Import window, open General, select Existing Projects into Workspace and click Next Figure 21. SW4STM32 workspace launcher dialog box
5. Click Select root directory, browse to the SW4STM32 workspace folder.
6. In the Projects panel, select the project and click Finish.
7. In the Project Explorer, select the project, open the Project menu, and click Build Project.
8. If the project is successfully compiled, the following messages display on the console window.
   To change the project settings (Include directories and preprocessor defines), simply go through Project>Properties, select C/C++ Build>Settings from the left panel:

   - For Include directories follow the path: C Compiler>Directories>Include path
   - For pre-processor defines follow the path: C Compiler>Symbols> Defined symbols

9. To debug and run the application, select the project in the Project Explorer and press F11 to start a debug session.

   The debugger in the AC6 SW4STM32 can be used to debug source code at the C and assembly levels, to set breakpoints, to monitor individual variables and to watch events during the code execution. To run an application select Resume, from the Run menu, or alternatively click the Resume button in the toolbar.

.8 GO TO PRODUCTION WITH CLOUD-JAM L4

In case the application has been developed with Nucleo functional pack it's possible to download directly the .bin file generated by the debug activity.
The best way to download the firmware using the ST-LINK V2 integrated on the Cloud-JAM L4 is to use the ST-LINK UTILY program.
It's possible to download it at this web address:

http://www.st.com/en/embedded-software/stsw-link004.html

.8.1 ST-LINK UTILITY PC REQUIREMENTS

The STM32 ST-LINK utility PC configuration requires as a minimum:

- PC with USB port and Intel® Pentium® processor running a 32-bit version of one of the following Microsoft® operating systems:
    - Windows® XP
    - Windows® 7
    - Windows® 10
- 256 Mbytes of RAM • 30 Mbytes of hard disk space available

.8.2 Installing the STM32 ST-LINK utility

Follow these steps and the on-screen instructions to install the STM32 ST-LINK utility:

1. Download the compressed STM32 ST-LINK utility software from the ST website.
2. Extract the contents of the .zip file into a temporary directory.
3. Double-click the extracted executable, setup.exe, to initiate the installation, and follow the on-screen prompts to install the STM32 ST-LINK utility in the development environment. The documentation for the utility is located in the subdirectory \Docs where the STM32 ST-LINK utility is installed.

Note:
If an earlier version of STM32 ST-LINK utility software is already installed, follow the uninstalling instructions described here in after before installing the new version.

.8.3 Uninstalling the STM32 ST-LINK utility

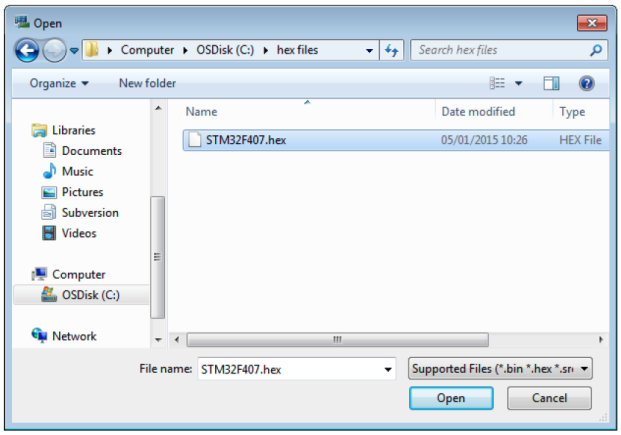
Follow these steps to uninstall the STM32 ST-LINK utility:

4. Select Start | Settings | Control Panel.
5. Double-click on Add or Remove Programs.
6. Select STM32 ST-LINK utility.
7. Click on the Remove button.

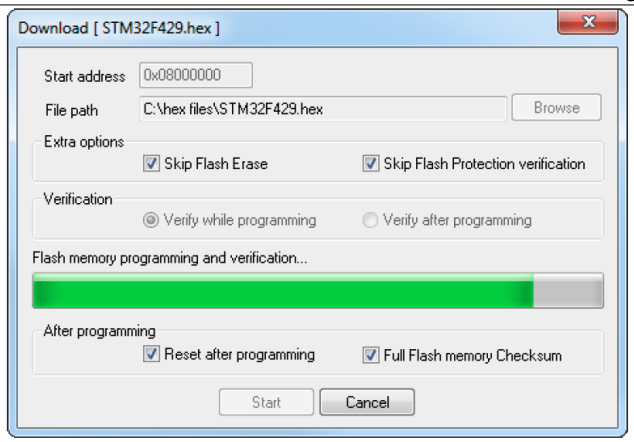
.8.4 Cloud-JAM L4 programming using ST-LINK UTILITY

The STM32 ST-LINK utility can download binary .bin, Hex, or srec files into Flash or RAM. To do this, follow these steps:

1. Click on Target | Program... (or Target | Program & Verify... if the user wants to verify the written data) to open the Open file dialog box, as shown in Figure below. If a binary file is already opened, go to step 3.



2. Select a binary, Intel Hex or Motorola S-record file and click on the Open button.
3. Specify the address from which to start programming, as shown in Figure 18: it may be a Flash or a RAM address.



4. Select Skip Flash erase option to skip flash erase operation in case the device is already erased
5. Select Skip Flash Protection verification to skip flash memory protection verification in case the device is not protected.
6. Choose a verification method by selecting one of the two radio buttons:
   6.1 Verify while programming: fast on-chip verification method which compares the program buffer content (portion of file) with the Flash memory content.
   6.2 Verify after programming: slow but reliable verification method which reads all the programmed memory zone after the program operation ends and compares it with the file content.
7. At last, click on the Start button to start programming:
   7.1 If Target | Program & Verify... is selected in the first step, a check is done during the programming operation.
   7.2 If the Reset after programming box is checked, an MCU reset will be issued.
8. Choose a verification method by selecting one of the two radio buttons:
   8.1 Verify while programming: fast on-chip verification method which compares the program buffer content (portion of file) with the Flash memory content.
   8.2 Verify after programming: slow but reliable verification method which reads all the programmed memory zone after the program operation ends and compares it with the file content.
9. At last, click on the Start button to start programming:
   9.1 If Target | Program & Verify... is selected in the first step, a check is done during the programming operation.
   9.2 If the Reset after programming box is checked, an MCU reset will be issued.

Note:1
The STM32F2 and STM32F4 Series supports different programming modes depending on the MCU supply voltage. When using ST-LINK, the MCU supply voltage should be specified in the Target | Settings Menu to be able to program the device with the correct mode. When using ST-LINK/V2, the supply voltage is detected automatically. If the device is read-protected, the protection will be disabled. If some Flash memory pages are write-protected, the protection will be disabled during programming and then recovered.

Note:2
The user can program Hex/Srec files that contains multiple segments for different target memory locations (Internal flash memory, external flash memory, Option bytes...). When programming the Read Out Protection to level 2 (debug and boot in SRAM/system Memory features are DISABLED), a message box will be displayed for confirmation to avoid protecting the chip by accident.

Note:3
The extra options are dedicated for programming operation on unprotected and erased devices.

Note:4
For any other detail please refer to UM0892 user manual of the STM32 ST-LINK Utility software that can be found here:
http://www.st.com/content/ccc/resource/technical/document/user_manual/e6/10/d8/80/d6/1d/4a/f2/CD00262073.pdf/files/CD00262073.pdf/jcr:content/translations/en.CD00262073.pdf

.9 TYPICAL APPLICATION

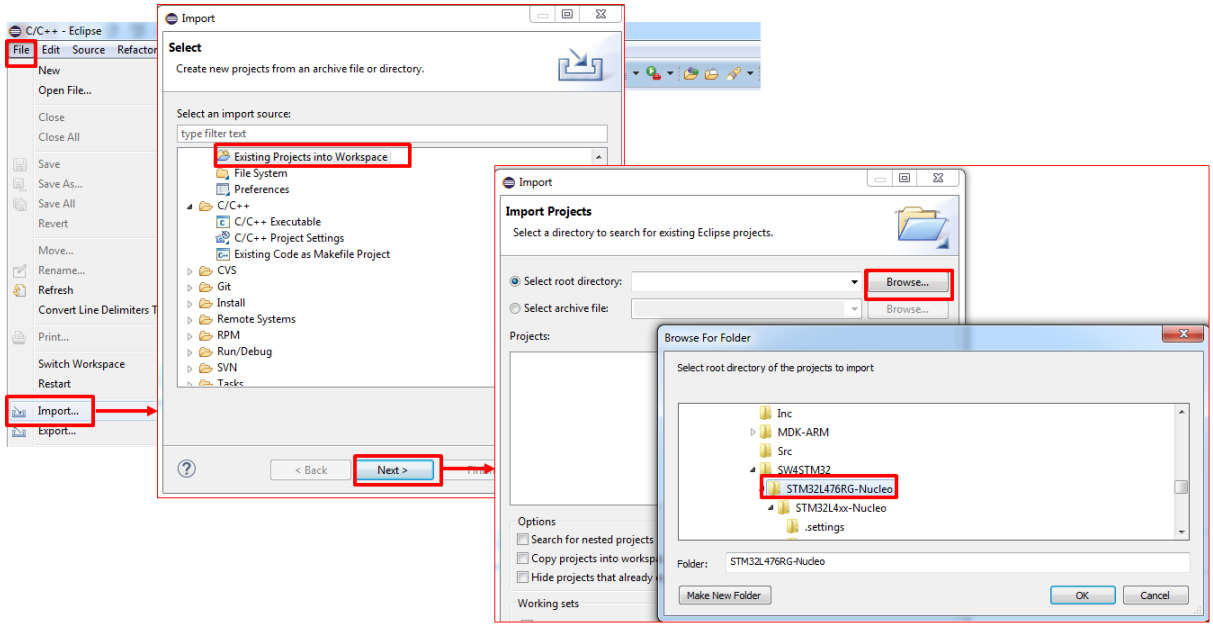
.9.1 FP-CLD-AZURE1 software description

Download FP-CLD-AZURE1 Function Pack.The Function Pack contains all the required drivers to use the Cloud-JAM L4 board with Wi-Fi, sensors and NFC expansion boards, together with pre-integrated Microsoft Azure IoT SDK.

Note: Use this software only for test. For production, download the last software of libraries from ST's website.

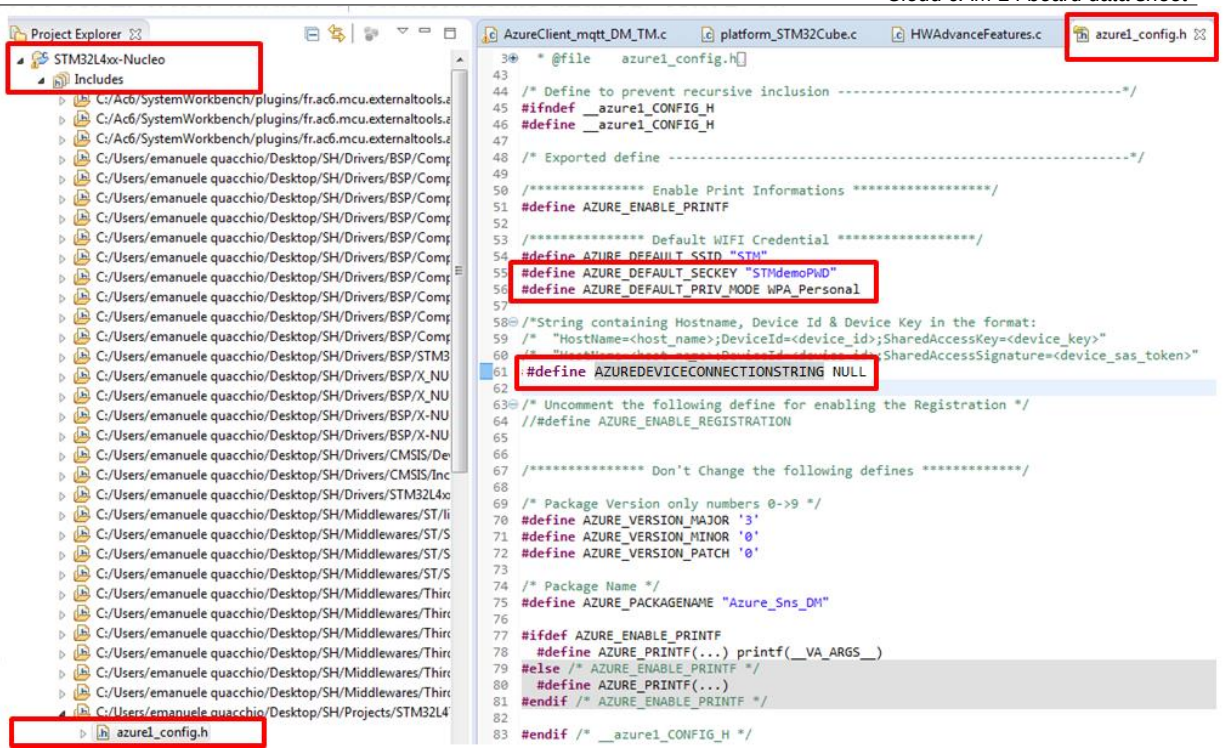
.9.1.1 Import solution file

Unzip the package and open one of the pre-configured project files available in Projects/STM32L476RG-Nucleo/Applications/Azure_Sns_DM, according to the IDE installed (for SystemWorkbench for STM32 project files can be found inside folder SW4STM32). In SystemWorkbench for STM32 select the project from menu File -> Import -> Existing Projects into Workspace; browse folders and select as root directory Projects/STM32L476RG-Nucleo/Applications/Azure_Sns_DM/SW4STM32/STM32L476RG-Nucleo then click Finish.



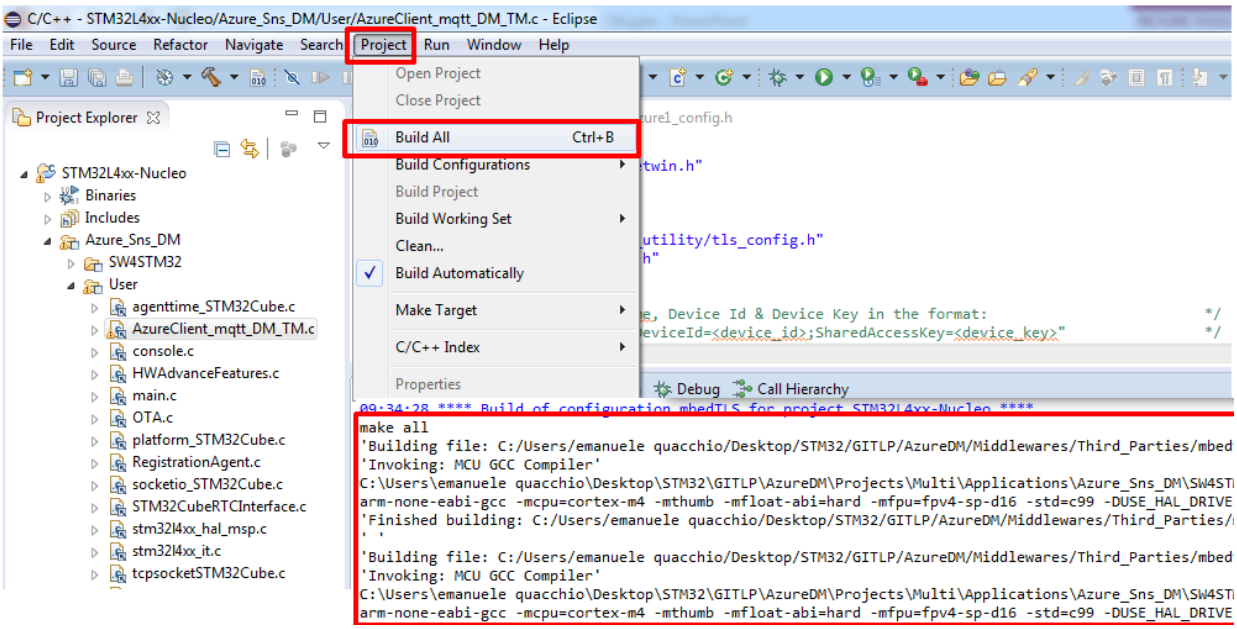
.9.1.2 Insert connection string

Open file azure1_config.h and update AZUREDEVICECONNECTIONSTRING with the credentials retrieved once completed device registration in IoT Hub as described in here. You have also to set here SSID and Password for Wi-Fi access point by replacing AZURE_DEFAULT_SSID and AZURE_DEFAULT_SECKEY.



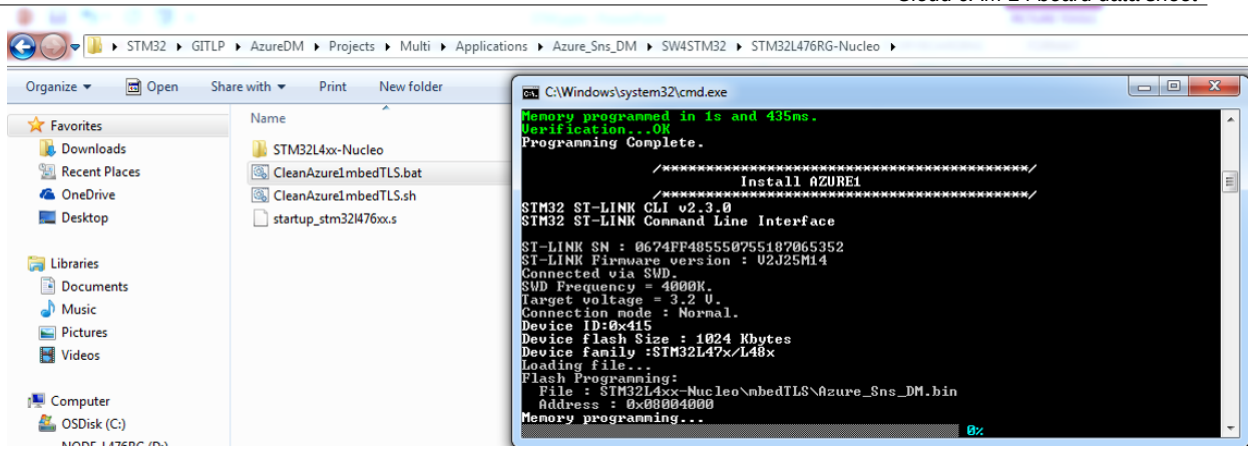
.9.1.3 Build the project

Build the project according to the selected IDE. In SystemWorkbench for STM32, click on Project -> Build All (or shortcut Ctrl+B).



.9.1.4 Flash the binary

Flash the binary to Cloud-JAM L4 board. The sample application you have compiled include a procedure to implement Over-The-Air (OTA) Firmware upadate, which can be combined with Azure IoT Hub primitives for device management (see following sections). Firmware update procedure requires a bootloader to be installed together with the Firmware binary; in order to properly flash both, scripts are provided for each IDE used. In SystemWorkbench for STM32 scripts for Windows/OSx/Linux can be found in Projects/STM32L476RG-Nucleo/Application/Azure_Sns_DM/SW4STM32/STM32L476RG-Nucleo. In Windows simply click on CleanAzure1mbedTLS.bat



For OSx/Linux, scripts require configuration according to your setup; edit OpenOCD_DIR and OpenOCD_CFC variables as shown in the following picture, following the installation path of openocd binaries (tools/openocd/) and scripts (openocd/scripts).

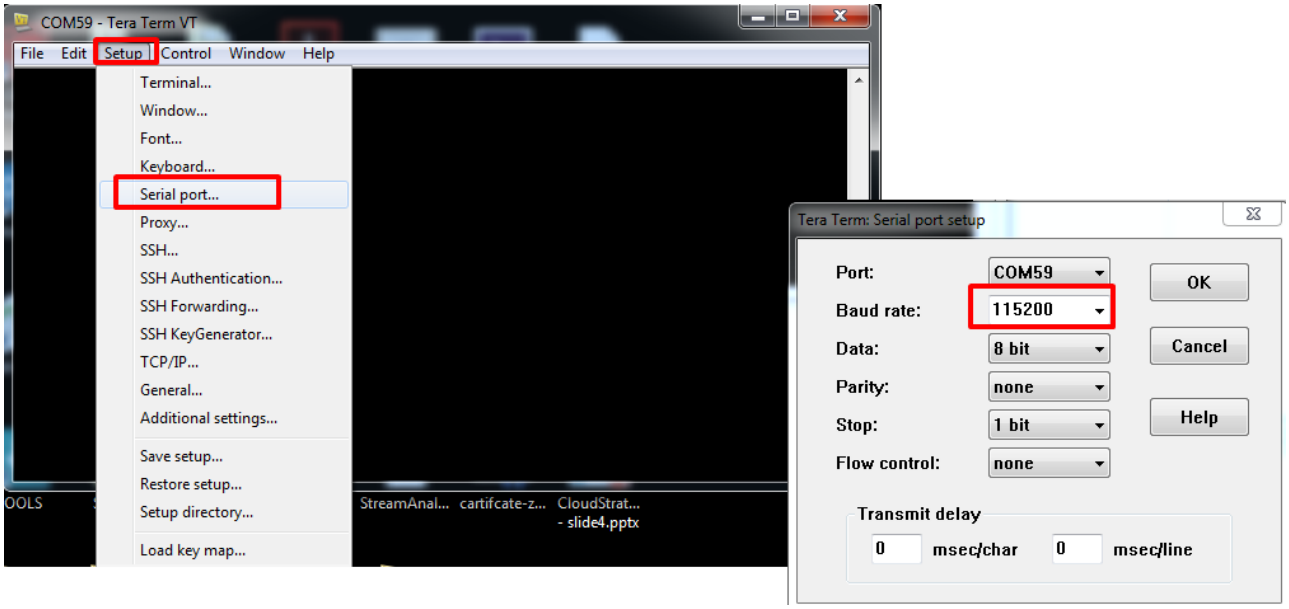
```
######## Modify this Section:
# 1) Set the Installation path for OpenOCD
# example:
OpenOCD_DIR="/Applications/Ac6/SystemWorkbench.app//Contents/Eclipse/plugins/
    fr.ac6.mcu.externaltools.openocd.macos64_1.11.0.201610101357/tools/openocd/"
#OpenOCD_DIR=""

# 2) Set the installation path for stm32 OpenOCD scritps
# example:
OpenOCD_CFC="/Applications/Ac6/SystemWorkbench.app//Contents/Eclipse/plugins/
    fr.ac6.mcu.debug_1.11.0.201610101240/resources/openocd/scripts"
#OpenOCD_CFC=""

# 3) Add openocd library path to _LIBRARY_PATH:
export DYLD_LIBRARY_PATH=${DYLD_LIBRARY_PATH}:${OpenOCD_DIR}"lib/"
```

.9.1.5 Configure your serial terminal and start the application

Configure your serial terminal as shown in the following picture for TeraTerm (baudrate set to 115200):



Reset the kit by pressing the RESET button on Cloud-JAM L4 board to start the application.

.9.1.6 Send Device Events to IoT Hub

After RESET, the application join the Wi-Fi access point and connect to your Azure IoT Hub. When connection is established, the application transmits periodically messages containing inertial and environmental data read from the board.

USER LED onboard blinks once connection with Azure IoT Hub is established. Application can be stopped by pressing USER button.

Messages successfully transmitted to your Azure IoT Hub are printed over your serial terminal interface.

```
Confirmation received for message [17] with result = IOTHUB_CLIENT_CONFIRMATION_SUCCESS
IoTHubClient_LL_SendEventAsync accepted message [18] for transmission to IoT Hub.
Confirmation received for message [18] with result = IOTHUB_CLIENT_CONFIRMATION_SUCCESS
IoTHubClient_LL_SendEventAsync accepted message [19] for transmission to IoT Hub.
Confirmation received for message [19] with result = IOTHUB_CLIENT_CONFIRMATION_SUCCESS
IoTHubClient_LL_SendEventAsync accepted message [20] for transmission to IoT Hub.
Confirmation received for message [20] with result = IOTHUB_CLIENT_CONFIRMATION_SUCCESS
IoTHubClient_LL_SendEventAsync accepted message [21] for transmission to IoT Hub.
Confirmation received for message [21] with result = IOTHUB_CLIENT_CONFIRMATION_SUCCESS
```

To visualize messages received in IoT Hub with iothub-explorer, open Node.js command prompt and insert the following commands:

```
iothub-explorer login <iot-hub-connection-string>

iothub-explorer monitor-events <device name> --login <iot-hub-connection-string>
```

```
C:\WINDOWS\system32\cmd.exe - iothub-explorer  monitor-events CloudJam1 --login HostName=CloudJamHub.azure-devices.net;SharedAccessKeyN...

C:\Users\matteo.fumagalli\Desktop>iothub-explorer monitor-events CloudJam1 --login Host
Name=CloudJamHub.azure-devices.net;SharedAccessKeyName=iothubowner;SharedAccessKey=5T2q

Monitoring events from device CloudJam1...
==== From: CloudJam1 ====
{
  "deviceId": "0080E1B7DB00",
  "messageId": 146,
  "temperature": 37.200001,
  "humidity": 33.200001,
  "accX": -16,
  "accY": -277,
  "accZ": 950,
  "gyrX": 3640,
  "gyrY": -3500,
  "gyrZ": -560,
  "ts": "2017-06-01T13:05:58Z"
}
---- properties ----
{
  "PropName": "PropMsg_146"
}
====================
==== From: CloudJam1 ====
{
```

.9.1.7 Receive messages from IoT Hub

To send a message from IoT Hub to Cloud-JAM L4 board with iothub-explorer, open Node.js command prompt and insert the following commands:

```
iothub-explorer send <device name> <message> --ack=full
```

Messages received by STM32 Cloud-JAM L4 are printed over serial terminal interface once received. Some cloud-to-device messages are also interpreted by the application:

- Pause : pause the application

- Play : restart the application after a pause
- LedOn/LedOff : turn on/off LED2 onboard Nucleo
- LedBlink : LED2 onboard Nucleo will blink for each message transmitted

See Manage IoT Hub to learn more on how to send cloud-to-device messages from IoT Hub.

.9.1.8 Trigger remote firmware update using Direct Methods and monitor device status with Reported Properties

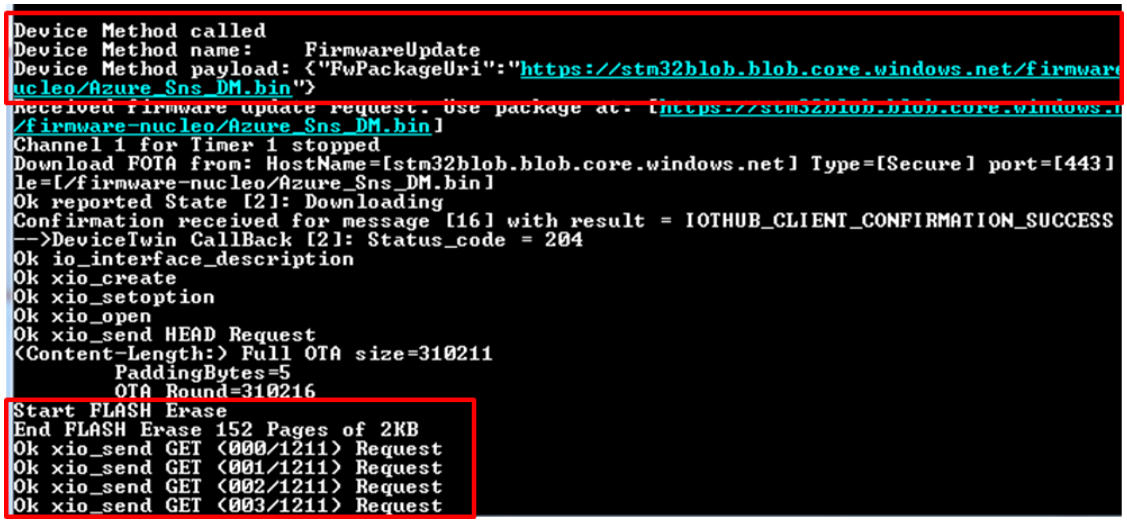
The application support firmware update procedure; when triggered, it stops its normal exection, download and install a new firmware version. By using direct methods, it is possible to trigger firmware update procedure using iothub-explorer; open Node.js command prompt and insert the following commands:

```
iothub-explorer device-method <device name> <method name> <method properties> <timeout>
```

where

- <method name> : FirmwareUpdate
- <method properties> : URL of the web-link where the new firmware version is hosted; to be written in the format "{\"FwPackageUri\":\"https:....\"}"
- <timeout> : timeout to receive feedback from device

It possible to monitor the execution of the firmware update procedure in the serial terminal:



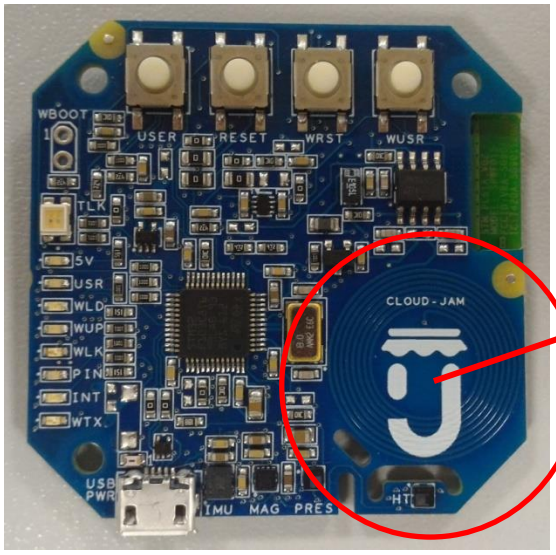
The application also reports its current status to the IoT Hub; device status can be monitored in iothub-explorer with the following command:

```
iothub-explorer get-twin <device name>
```

```
C:\Users>iothub-explorer get-twin P-NUCLEO-AZURE1-EQ
deviceId:    P-NUCLEO-AZURE1-EQ
etag:        AAAAAAAAAAY=
properties:
  desired:
    DesiredTelemetryInterval: 4
    $metadata:
      $lastUpdated:                 2017-04-27T11:27:06.0083642Z
      $lastUpdatedVersion:          6
      DesiredTelemetryInterval:
        $lastUpdated:           2017-04-27T11:27:06.0083642Z
        $lastUpdatedVersion: 6
    $version:                       6
  reported:
    AzureFwVersion:     Azure_Sns_DM V3.0.0 SDK=1.1.6
    AzureStatus:        Downloading
    TelemetryInterval:
    SupportedMethods:
      Reboot:                                   Reboot the device
      Quit:                                     Stop the trasmission
      FirmwareUpdate--FwPackageUri-string: Updates device Firmware.
                    Use parameter FwPackageUri to specifiy the URI of the firm
ware file
```

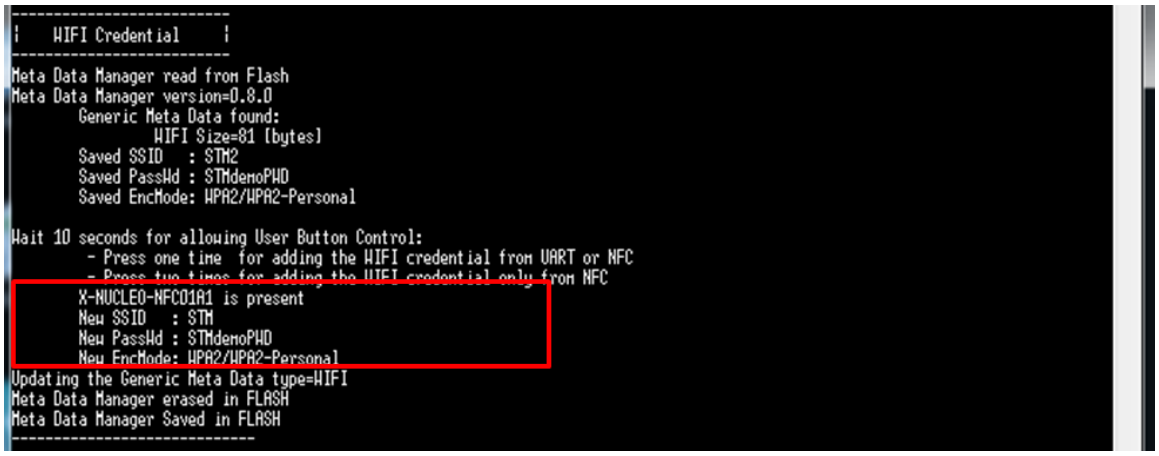
.9.2 Use NFC to configure Wi-Fi Access Point parameters

- Install in an Android phone ST25 NFC App mobile app
- Launch the application; click on Compose NDEF then select in menu the Wi-Fi option. Insert SSID and Password, then approach the mobile phone to the NFC expansion board and click on Write to tag



NFC ANTENNA AREA

- Press Reset Button to restart the application
- Press twice the User Button (Blue Button) when requested (after 3 seconds timeout default values will be used). The application will read NDEF parameters from NFC and will connect to the Access Point.

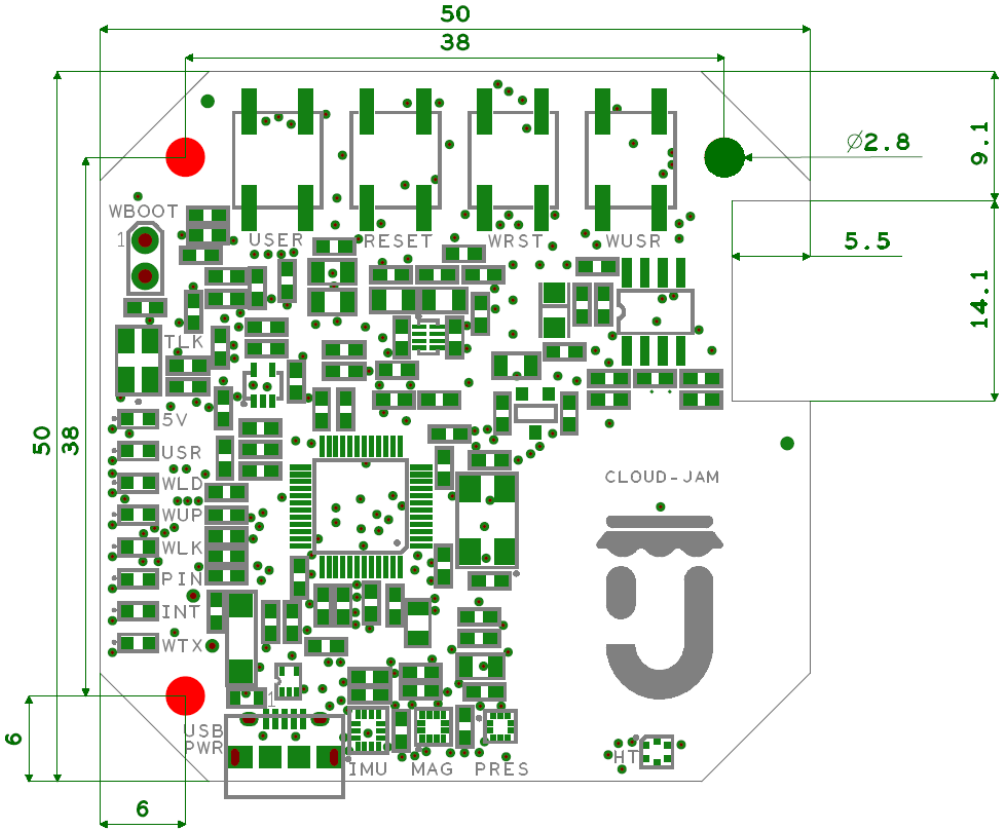


.10 ELECTRICAL SPECIFICATIONS

Parameter	Minimum	Typical	Maximum	Conditions
Power supply voltage	3,5V	5V	5,5V	USB port input connector
Input current	-	-	0.5A	Protected by resettable PTC
Operation temperature	-20°C	-	70°C	-
Storage temperature	-40°C		85°C	-

Wireless standard	Frequency band low	Frequency band high	Transmitted power
Wi-Fi	2402 MHz	2480 MHz	11 dBm
NFC	13.56MHz	13,56 MHz	Passive NFC

.11 MECHANICAL SPECIFICATIONS



The measures are in mm and the view is TOP side.

.12 HOUSING PRESCRIPTION

Cloud-JAM L4 is PCBA (Printed Circuit Board Assembly) system and then is not a final product, it's necessary to close the board in a specific housing or mechanical box.
The housing, in addition to achieve the desired functionality, musts be fireproof (in case of unexpected failure of the electronic board component that produce a little spark, a fire will be blocked).

.13 PACKAGING

Packaging contain Cloud-JAM L4 board and the USB cable.

.14 UE DECLARATION OF CONFORMITY

EU Declaration of Conformity (DoC)

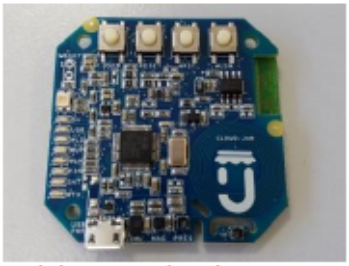
We

Company name:	RushUp S.r.l.
Postal address:	C. Battisti 136
Postcode:	24025
City:	Gazzaniga (Bg)
Telephone number:	+3903573180
E-Mail address:	info@rushup.tech

declare that the DoC is issued under our sole responsibility and belongs to the following product:

Apparatus model/Product:	Cloud-JAM L4
Type:	STM32ODE IoT electronics board
Batch:	CJL4
Serial number:	From 000001 to 999999

Object of the declaration (identification of apparatus allowing traceability; it may include a colour image of sufficient clarity where necessary for the identification of the apparatus):



The object of the declaration described above is in conformity with the relevant Union legislation:

Directive 2014/53/EU (RED)

RED Directive 2014/53/EU	
RoHS Directive 2011/65/EU	

The following standards and technical specifications have been applied:

Title, Date of standard/specification:

EN 62311:2008	ETSI EN 301 489-17 V3.2.0
ETSI EN 301 489-1 V2.2.0	ETSI EN 300 328 V2.1.1
ETSI EN 301 489-3 V2.1.1	

Notified body (where applicable):	4 digit notified body number:
Nemko S.p.A.	2051

Reference number of the certificate of notified body: 2051-RED-173801

Additional information:

Signed for and on behalf of:

Gazzaniga (Bg)	2017/10/31	Christian Raineri, CEO
Place of issue	Date of issue	Name, function, signature

.15 OPERATING ENVIRONMENT

The operating environment excludes special environments (extreme temperatures, dust, humidity, vibrations, flammable gases, corrosive or explosive atmosphere, etc.).

.16 DICLAIMERS

Products and specifications discussed herein are for reference purposes only. All information discussed herein is provided on an "AS IS" basis, without warranties of any kind. This document and all information discussed herein remain the sole and exclusive property of RUSHUP. No license of any patent, copyright, mask work, trademark or any other intellectual property right is granted by one party to the other party under this document, by implication, estoppel or other-wise.
RUSHUP products are not intended for use in life support, critical care, medical, safety equipment, or similar applications where product failure could result in loss of life or personal or physical harm, or any military or defense application, or any governmental procurement to which special terms or provisions may apply. For updates or additional information about RUSHUP products, contact RUSHUP office. All brand names, trademarks and registered trademarks belong to their respective owners.
RUSHUP and JAM are trademarks of RUSHUP srl.