

Examples	2
1. Node Red dashboard controlling Led Strips (Arduino)	2
2. Node Red dashboard reading Environment sensor (Nucleo board)	2
3. Push Button(Nucleo Board) changing color of LED Driver 3 click (Nucleo Board)	3
FT Click	3
Wiring FT click Transceivers	3
Setup	4
Arduino	4
Development Environment	4
Wiring	6
Test the LED strips	7
Demo code	7
Connect FT Wiring	8
STmicro	8
Development Environment	8
Assembly and Wiring	8
Nucleo boards	8
Connect FT Wiring	12
Raspberry Pi	12
What You Will Need to Download	12
Assembly	13
Setup	13
Install Node Red on RPI 3B+	13
Configure node-red	14
Run simulated data generator	18
Set-up and run ftmq_mqtt_bridge	18
MQTT demo	19
MQL plugged in	19
Connect FT Wiring	19
Test	20
Resources	20
Contact	20

Examples

This Document describes the steps needed to assembly the FTclick examples. See the [Resources section](#) to get the repository with the examples and libs. If you have any trouble, check the [Contact section](#).

There are 3 main examples:

1. Node Red dashboard controlling Led Strips (Arduino)

In this example we will communicate a Raspberry Pi with an Arduino UNO using the FTclick. This will allow us to control two led strips attached to an arduino uno using a Node Red Dashboard

There are several steps needed to make it work:

1. Check the Examples BOM document and buy all you need for this example.
2. Assembly and configure the Arduino UNO with the LED strips. [Explained Here](#)
3. Assembly and configure the Raspberry pi. [Explained Here](#)
4. Wire the connections using the connectors that come with the FTclick. [Explained Here](#)
5. Connect everything.

2. Node Red dashboard reading Environment sensor (Nucleo board)

In this example we will communicate a Raspberry Pi with a Nucleo Board using the Ftclick to show the readings of an environment sensor in a Node Red Dashboard.

There are several steps needed to make it work:

1. Check the Examples BOM document and buy all you need for this example
2. Assembly and configure the Nucleo Board with the Environment sensor. [Explained Here](#)
3. Assembly and configure the Raspberry pi. [Explained Here](#)
4. Wire the connections using the connectors that come with the FTclick. [Explained Here](#)
5. Connect everything.

3. Push Button(Nucleo Board) changing color of LED Driver 3 click (Nucleo Board)

In this example we will communicate two Nucleo Boards using the FTclick. When the button is pressed the Led in the other Nucleo should change colour.

There are several steps needed to make it work:

1. Check the Examples BOM document and buy all you need for this example
2. Assembly and configure the Nucleo Board with the Push Button. [Explained Here](#)
3. Assembly and configure the Nucleo Board with the LED Driver 3. [Explained Here](#)
4. Wire the connections using the connectors that come with the FTclick. [Explained Here](#)
5. Connect everything.

FT Click

Wiring FT click Transceivers

FT (Free Topology) is an industry-wide adopted field network connectivity standard as specified in ISO/IEC 14908-2 (also known as the TP/FT-10 channel type), that is:

- Free of network topology limitations
 - Supports Bus, Star, Loop or any topology
 - Easy to expand
- Free of installation hassles
 - Polarity insensitive, two wires only
 - Single termination (in free topology)
- Free of errors
 - High noise immunity
 - Fault-tolerant, simple error diagnostics
- Free of high cabling costs
 - Wide choice of unshielded twisted-pair or Cat 5 cable

Each of the FT click boards should be connected together using a suitable twisted pair cable and terminated with a single 52.3W 5% 0.25W resistor for free topology support. FT click boards can be wired in a simple daisy chain basis thanks to the convenient connectors that require no tools.

Five cable types have been validated for the TP/FT-10 channel. These cable types are:

- TIA 568A Category 5 cable, 24AWG, 0.5mm
- Belden 8471 (PVC jacket) or equivalent cable, 16AWG, 1.3mm
- Belden 85102 (Tefzel jacket) or equal cable, 16AWG, 1.3mm
- Level IV cable, 22AWG, 0.65mm
- JY (st) Y 2x2x0.8, 20.4AWG, 0.8mm

Setup

The FTClick has built-in commands that allows you to configure it in some special cases and to read software and hardware versions. To do this you will have to connect the FT Click tx and rx pins to a serial adaptor. Some examples are explained below.

Set the node id (x is the nodeid that we want to set):

```
"python ftclick_commands.py COM node-id set x"
```

Get the node id:

```
"python ftclick_commands.py COM node-id get"
```

Get the hardware version:

```
"python ftclick_commands.py COM hw-ver"
```

Get the software version:

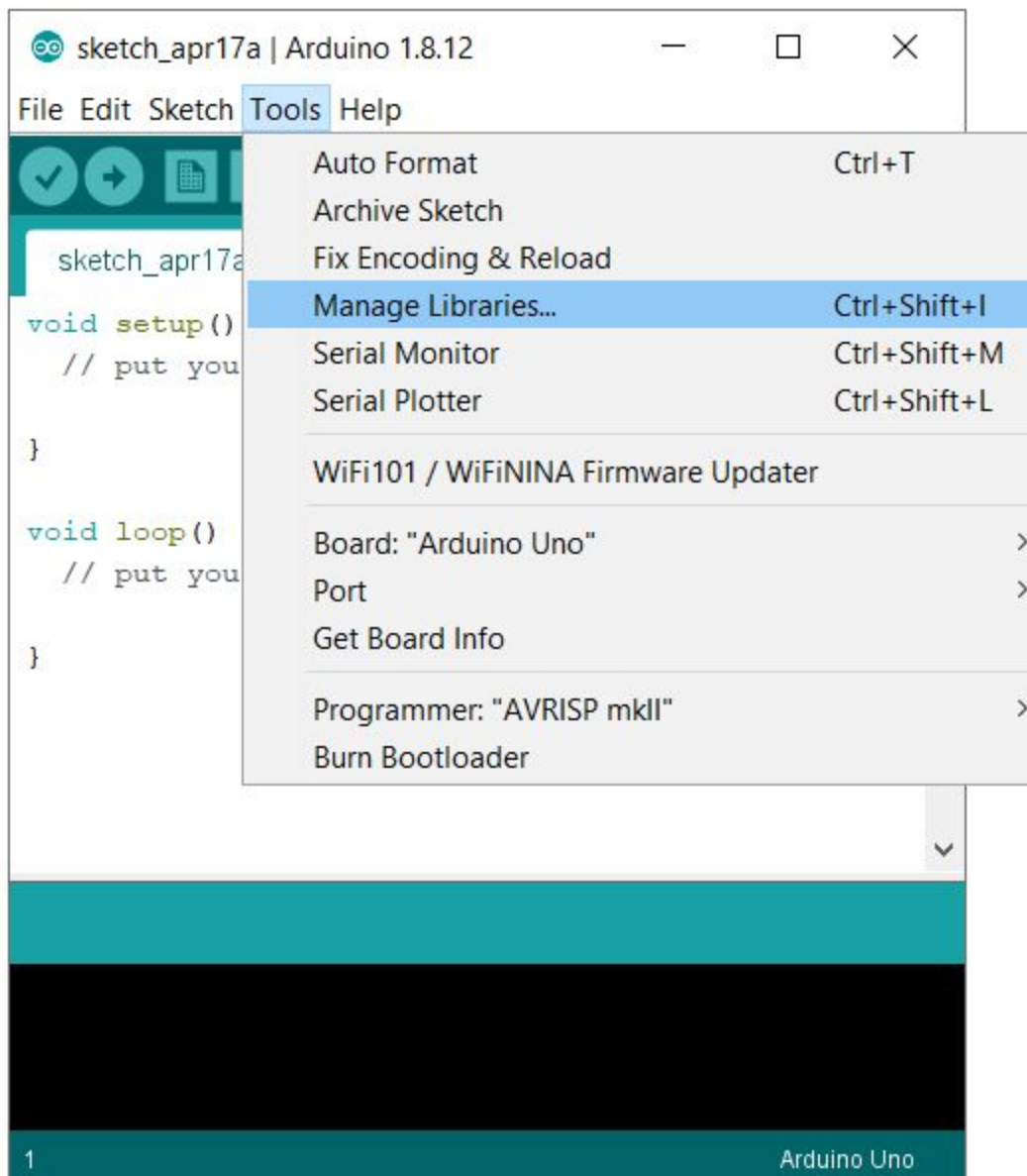
```
"python ftclick_commands.py COM sw-ver"
```

Arduino

Development Environment

Arduino IDE

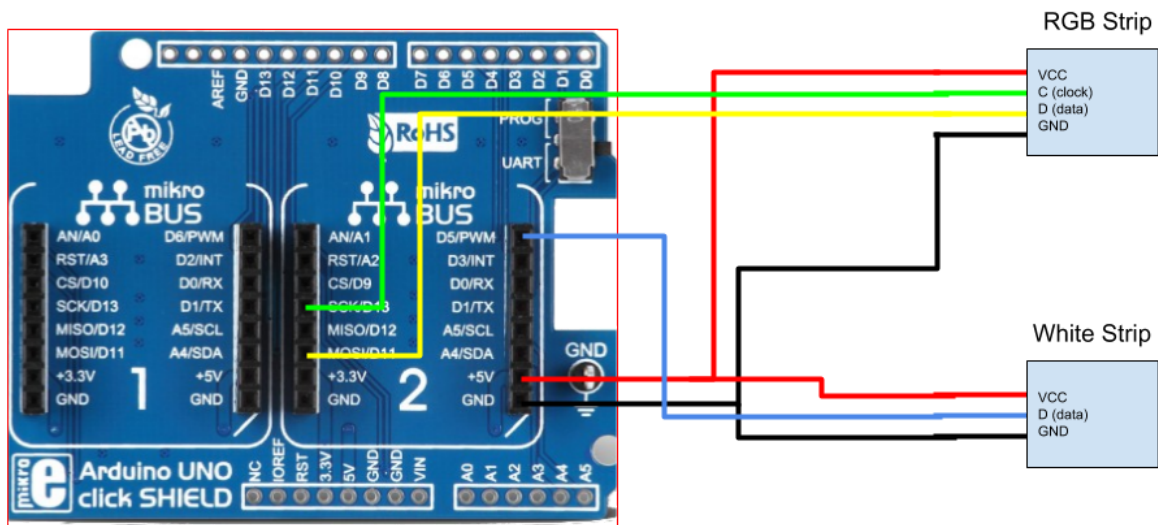
- Clone the software repo "Usrlibs repo" (be sure to have the "master" branch checked out) with the command "git clone"
- Download and install arduino IDE v1.8.11 or newer (not the cloud version)
<https://www.arduino.cc/en/Main/Software>
- Once installed, open arduino ide and go to the library manager (tools-> Library Manager)



- <https://www.arduino.cc/en/guide/libraries>
- Search and install the following libraries:
 - Adafruit Neopixel (1.3.5)
 - Adafruit DotStar (1.1.4)
 - ArduinoJson 6.15.1 by Benoit Blanchon
- Copy ftmq and ccp library in the Arduino libraries folder (in windows Documents\Arduino\libraries) (in linux ~/Arduino/libraries) from your_platform/ftmq and your_platform/ccp
- Restart the arduino IDE to make sure that all changes are made and can work with the libs now

Wiring

You should see an arrow going through the led strip from one side to other, the next diagram should be used with the side of the led strip that has the arrow leaving or not pointing.



LED Strip	Micro BUS Pin	LED Pin
RGB	MOSI/D11	D (Data)
RGB	SCK/D13	C (Clock)
White	D5/PWM	D (Data)

NOTE: The arduino click shield has a 2-position switch. To program the arduino with the shield mounted, the switch has to be in “PROG” position. This position allows the arduino to be programmed from the computer through the usb port, but prevents an FT click module from communicating with the arduino. For normal operation, the switch has to be in “UART” position, allowing the communications between the arduino and the FT click.

An external 5V PSU for the led strips is recommended with at least 2A current.

Test the LED strips

Load the sketch “ewdemo_test_strips.ino” into the arduino.

https://bitbucket.org/connectexteam/userlibs/src/master/Arduino/Arduino_Uno/examples/ewdemo_test_strips/

The program will turn on led on both strips, cycling the three base colors, and performing a shift along the strips. In the beginning of the sketch source the length of both strips has to be defined (number of leds)

```
#include <Adafruit_DotStar.h>
// Because conditional #includes don't work w/Arduino sketches...
#include <SPI.h>          // COMMENT OUT THIS LINE FOR GEMMA OR TRINKET
//#include <avr/power.h> // ENABLE THIS LINE FOR GEMMA OR TRINKET

#define LED_COUNT 144 // Number of LEDs in White strip

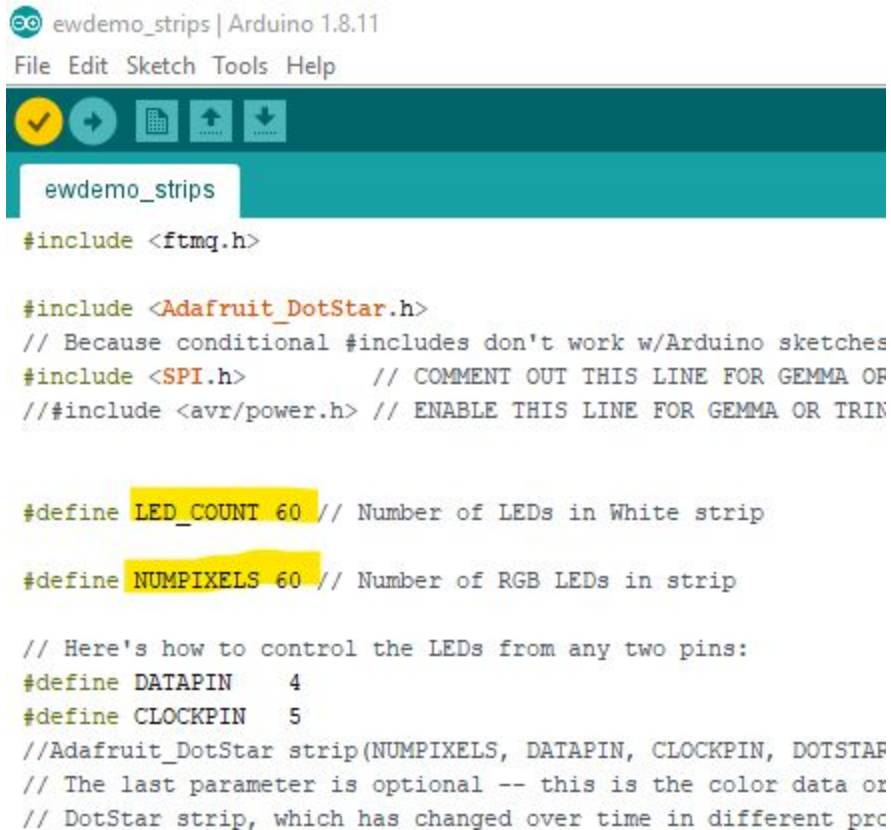
#define NUMPIXELS 144 // Number of LEDs in RGB strip
```

Demo code

Load the sketch “ewdemo_strips.ino” into the arduino.

https://bitbucket.org/connectexteam/userlibs/src/master/Arduino/Arduino_Uno/examples/ewdemo_strips/

The strip led count has to be adjusted in the source:



```
ewdemo_strips | Arduino 1.8.11
File Edit Sketch Tools Help

ewdemo_strips

#include <ftmq.h>

#include <Adafruit_DotStar.h>
// Because conditional #includes don't work w/Arduino sketches
#include <SPI.h>          // COMMENT OUT THIS LINE FOR GEMMA OF
//#include <avr/power.h> // ENABLE THIS LINE FOR GEMMA OR TRIN

#define LED_COUNT 60 // Number of LEDs in White strip

#define NUMPIXELS 60 // Number of RGB LEDs in strip

// Here's how to control the LEDs from any two pins:
#define DATAPIN 4
#define CLOCKPIN 5
//Adafruit_DotStar strip(NUMPIXELS, DATAPIN, CLOCKPIN, DOTSTAF
// The last parameter is optional -- this is the color data or
// DotStar strip, which has changed over time in different prc
```

Usually 1 meter has 144 LEDs in the high density strips.

Connect FT Wiring

Connect the FT twisted pair wiring to the FT click board.

STmicro

Development Environment

STM32CubeIDE

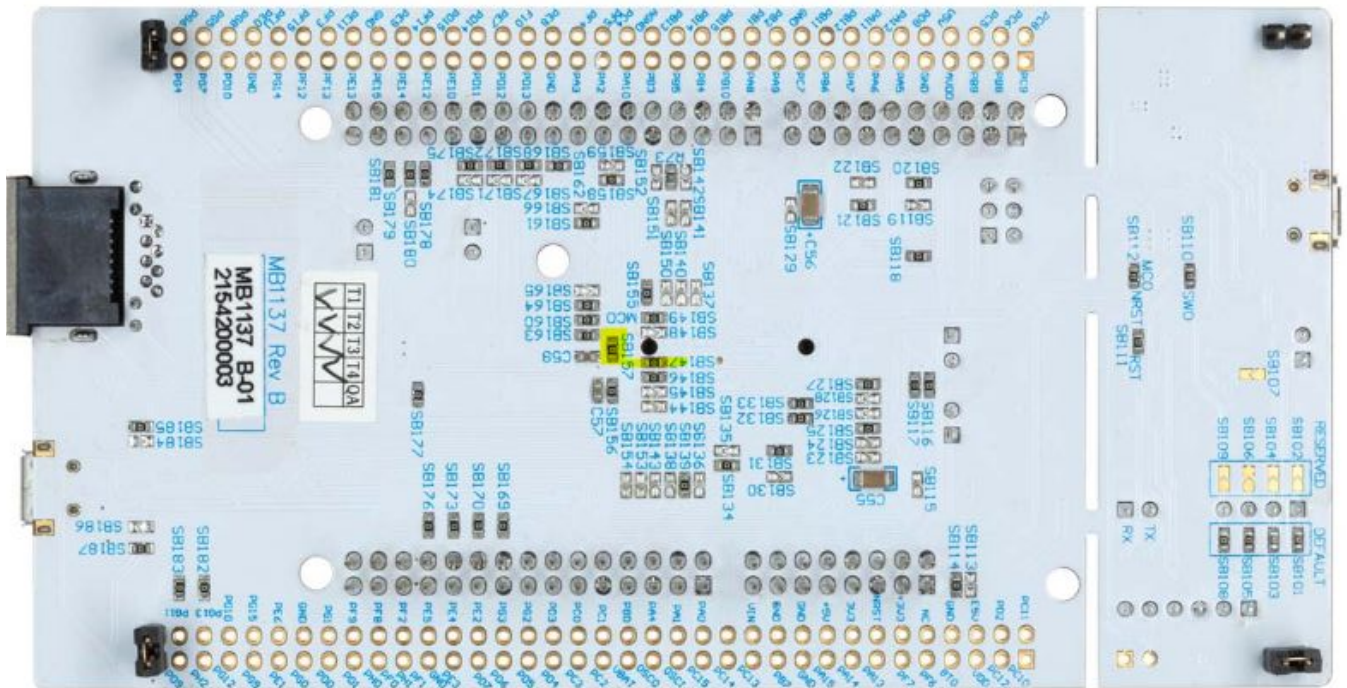
- Download and install STM32CubeIDE v1.2.0
<https://www.st.com/en/development-tools/stm32cubeide.html#get-software>

Assembly and Wiring

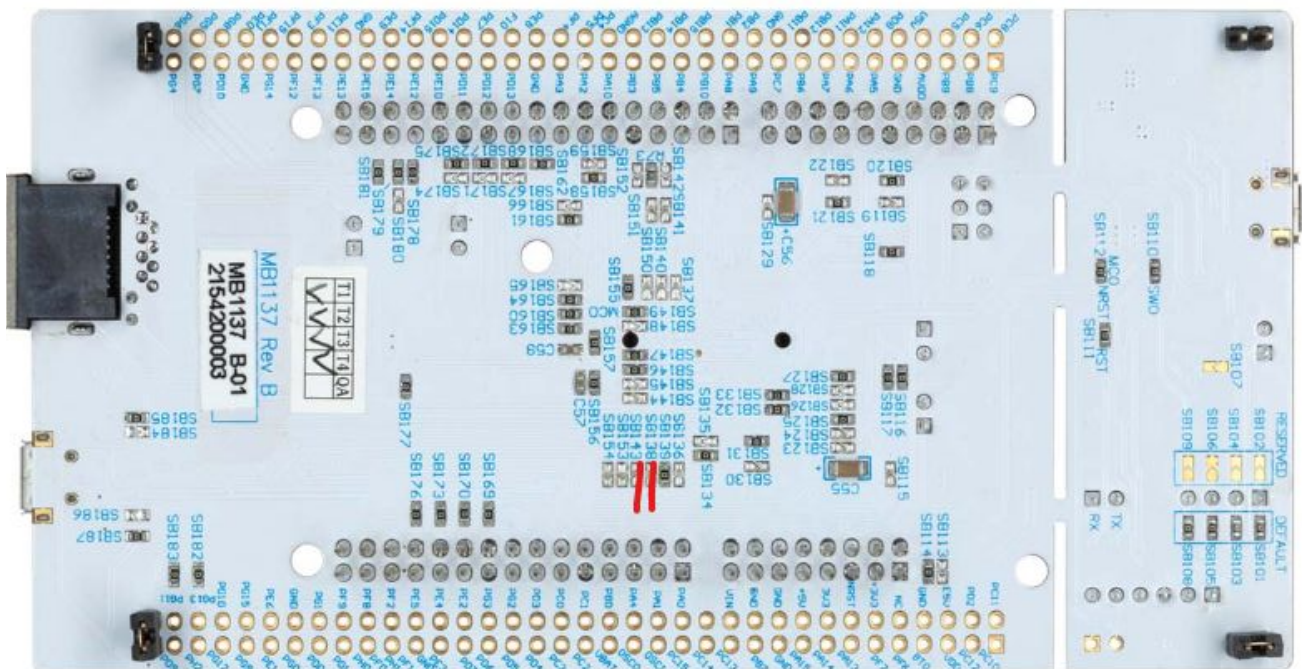
Nucleo boards

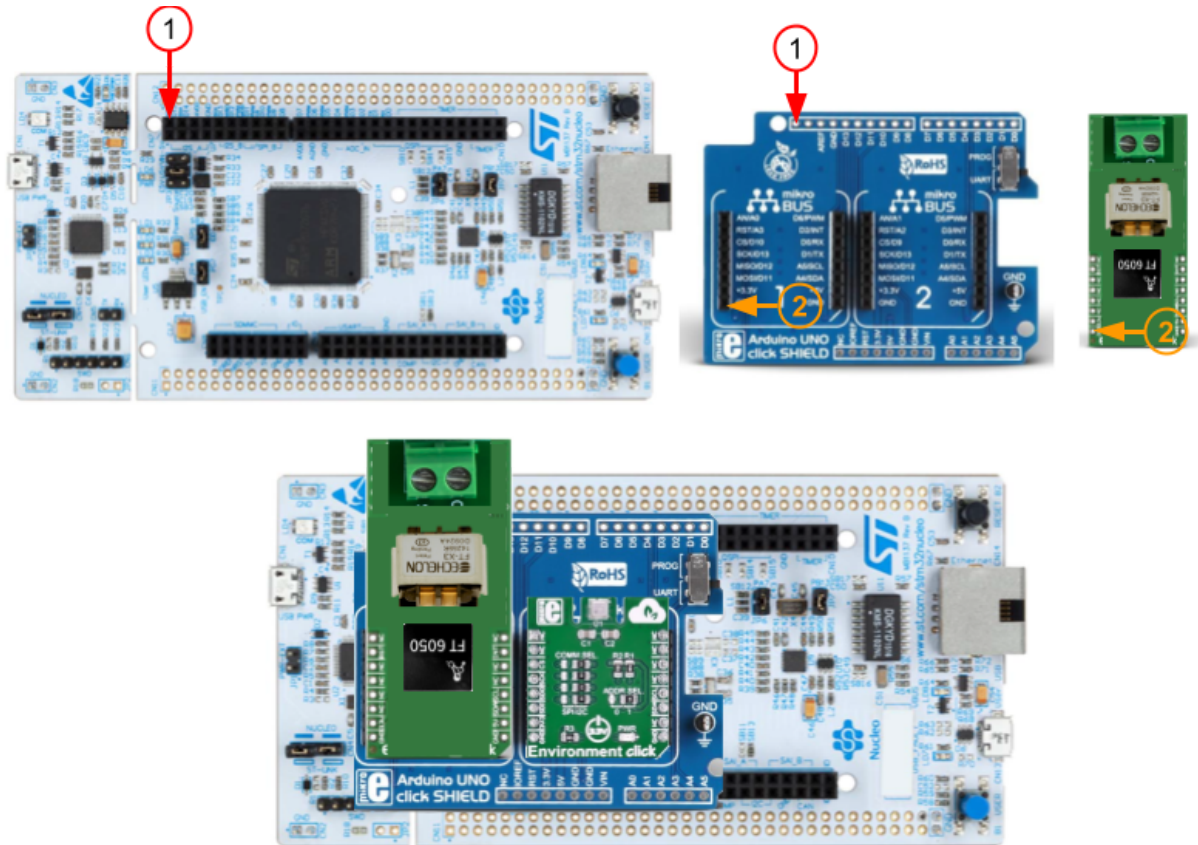
The F429ZI nucleo boards need a set of changes in the solder jumpers in order to remap I2c to the arduino header. The can be connected with solder or with 0 ohm smd resistors.

1. Remove the marked jumpers (R0 ohm smd resistors):

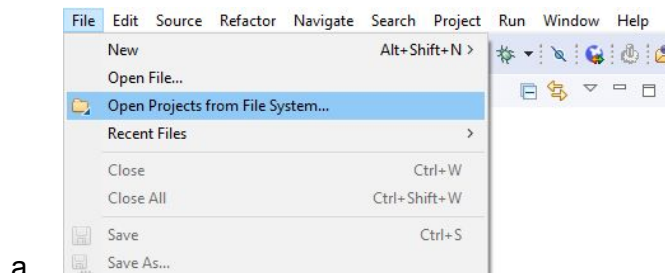


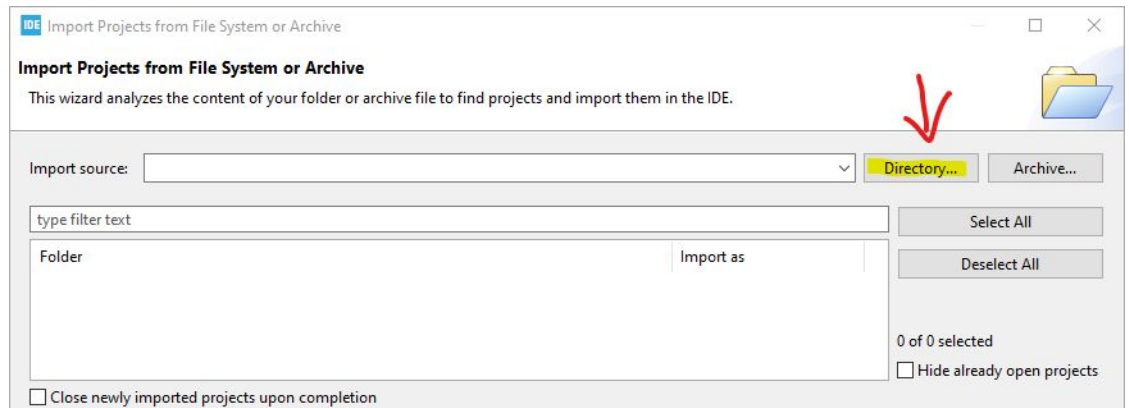
2. Connect the marked solder jumpers:



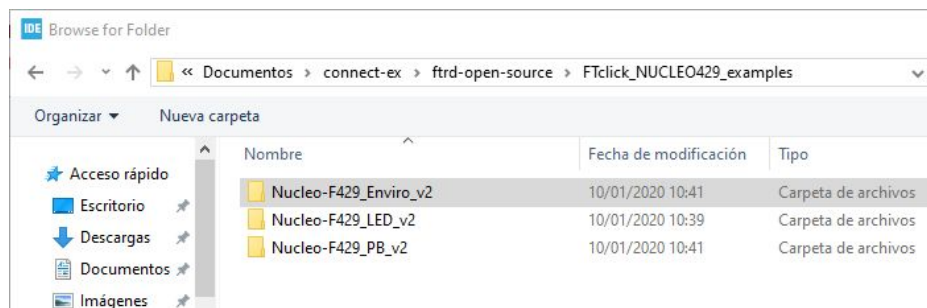


1. Connect adapter arduino to Nucleo following the image.
2. On top of this adapter, connect the FTclick, also as the previous image shows.
3. Connect the BME680 click, the click LED Driver 3 or the Pushbutton, depending on the example that you are building, on the available socket.
4. Download source code from repository:
 - [BME680 example](#)
 - [PushButton example](#)
 - [click LED Driver 3](#)
5. Import project into STM32CubeIDE

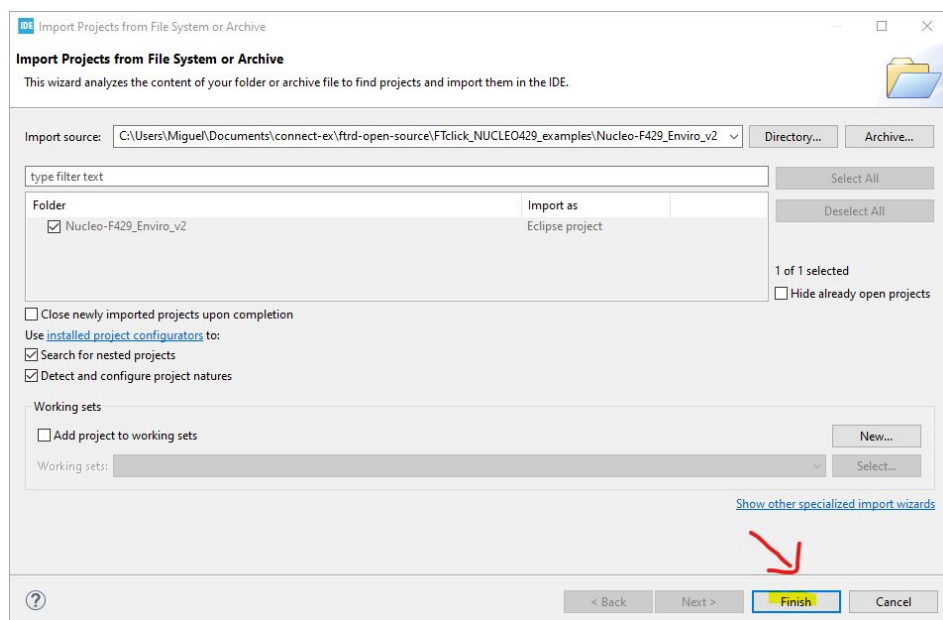




b.



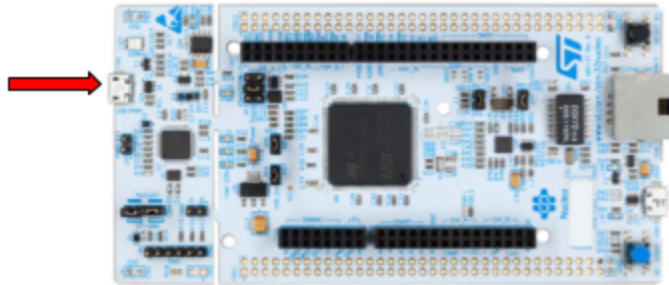
c.



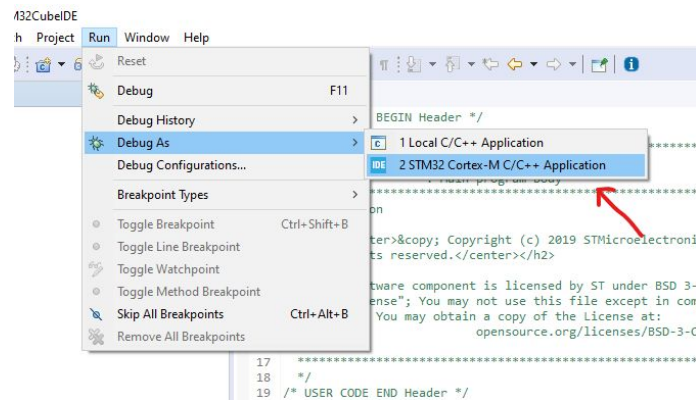
d.

6. Connect Nucleo board usb port to PC

USB port for
power and
programming



7. Click on “Debug” to upload the code to the board



8. Stop the debugging session and unplug the usb

Connect FT Wiring

Connect the FT twisted pair wiring to the FT click board.

Raspberry Pi

What You Will Need to Download

- Win32diskimager from <https://sourceforge.net/projects/win32diskimager/> or balenaetcher from <https://www.balena.io/etcher/>
- An SSH client such as PuTTY from <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>

Assembly



1. Connect Click Hat
2. On top of this adapter, connect the FTclick, also as the previous image shows.

Setup

Install Node Red on RPi 3B+

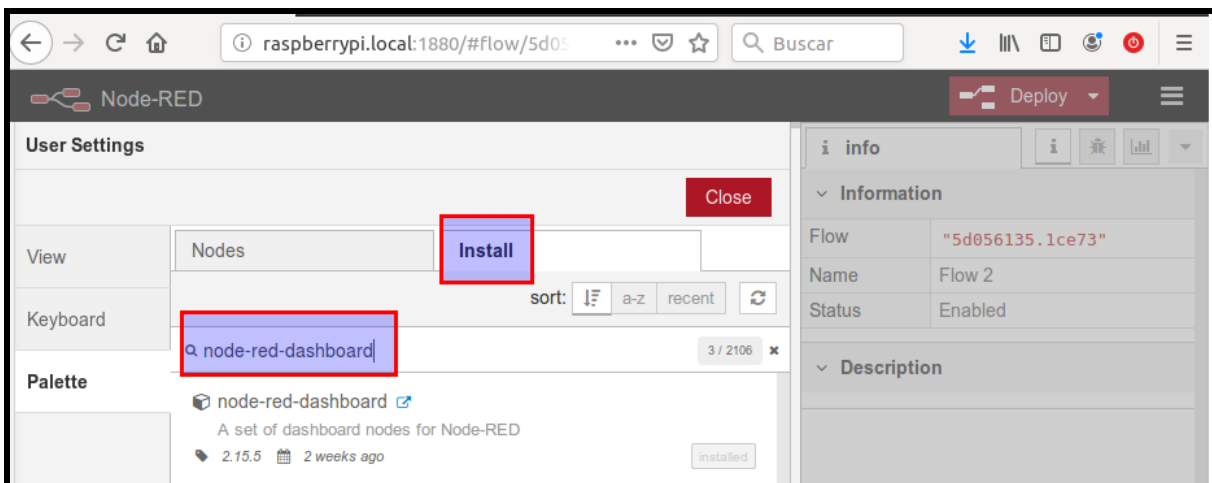
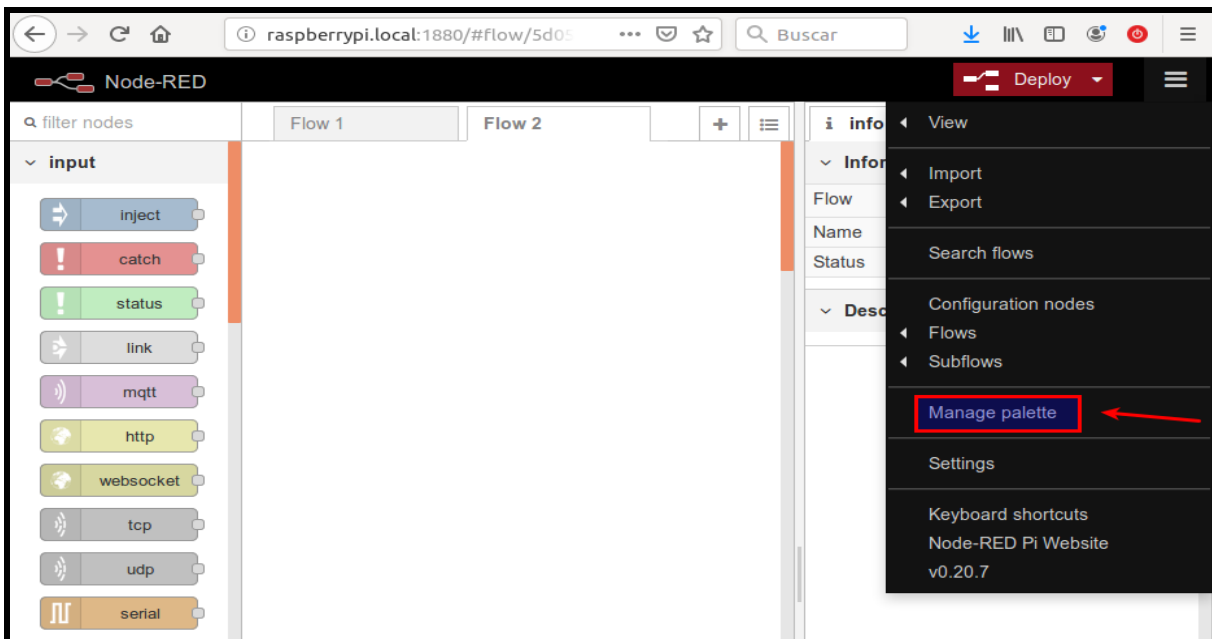
- Download Image (raspbian buster has a bug with HDMI on 3B+)
 - https://downloads.raspberrypi.org/raspbian_lite/images/raspbian_lite-2019-04-09/
- Burn the image with Balena Etcher on the SD card
- Power on the RPi and enable SSH:
 - `sudo raspi-config`
 - Select Interface Options
 - Select SSH
 - Confirm SSH
- Update and Upgrade the OS
 - `sudo apt-get update`
 - `sudo apt-get upgrade`
- Install some requirements
 - `sudo apt-get install git`
 - `sudo apt-get install build-essential`
- Install the broker
 - `sudo apt-get install mosquitto mosquitto-clients`
- Install node-red
 - `bash <(curl -sL https://raw.githubusercontent.com/node-red/linux-installer/s/master/deb/update-nodejs-and-nodered)`
- Update and Upgrade the OS again
 - `sudo apt-get update`
 - `sudo apt-get upgrade`
- Run node-red as service on boot
 - `sudo systemctl enable nodered.service`
- Restart the node-red
 - `sudo systemctl restart nodered.service`

- Clone the repo
 - `sudo git clone git clone https://bitbucket.org/connectexteam/userlibs.git`
- Reboot the RPi
 - `sudo reboot`

Configure node-red

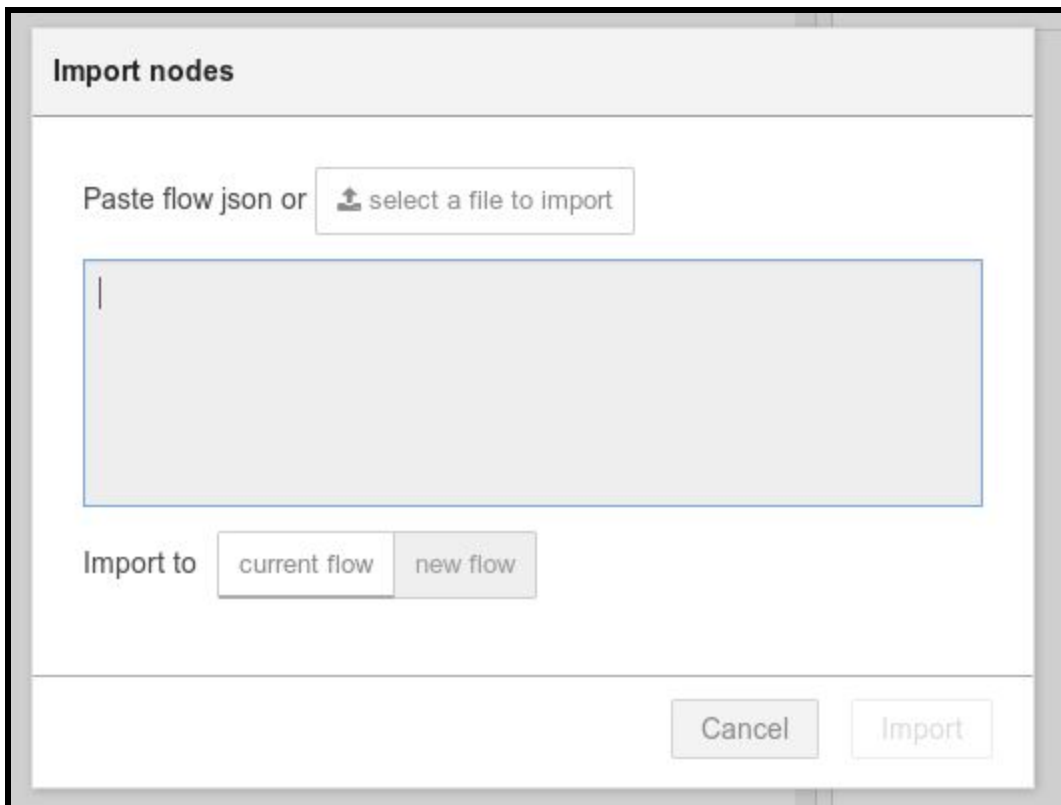
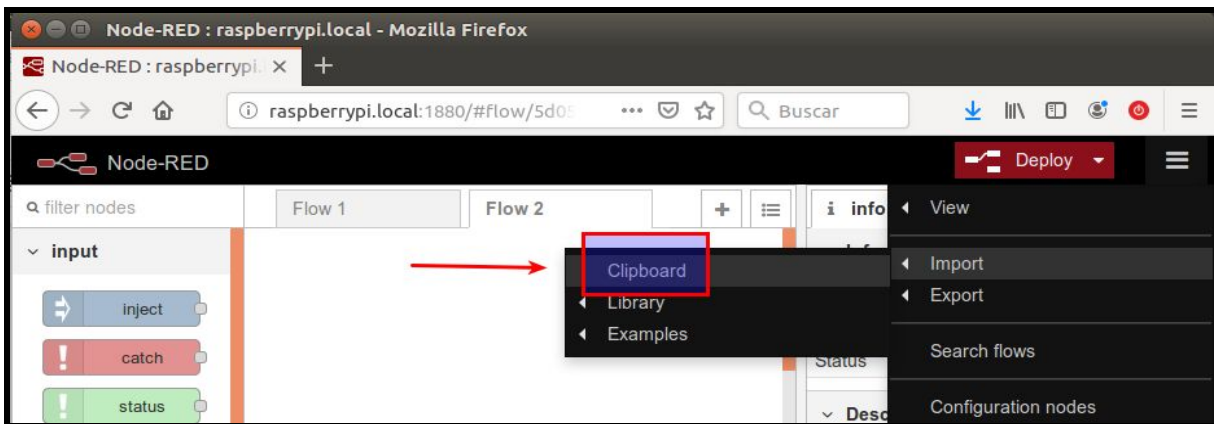
Now you can access the web panel (*raspberrypi-ip:1880*)

- On node-red dashboard, go to “Manage Palette” in the menu, Switch to “Install”, search and install **node-red-dashboard** and **node-red-contrib-ui-led**.

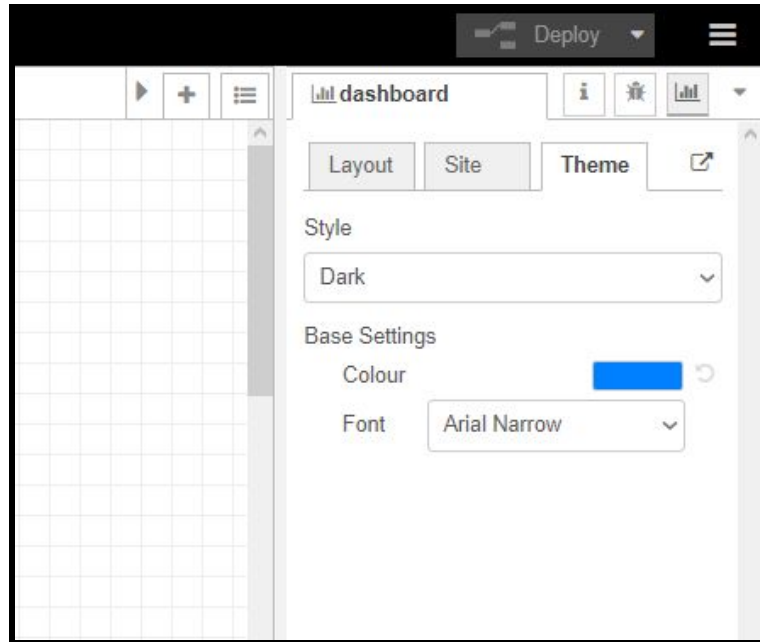


- Restart the node-red on RPi
 - `sudo systemctl restart nodered.service`
- Import node-red flow

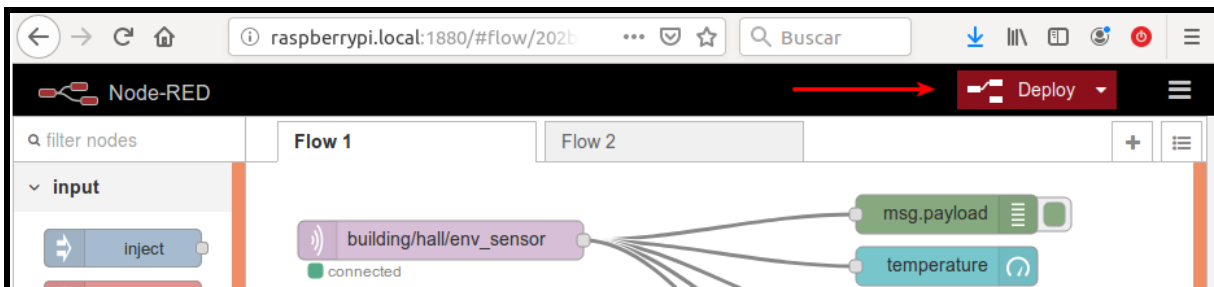
- https://bitbucket.org/connectexteam/userlibs/src/master/Raspberry_Pi/node-red-flows/flows.json)



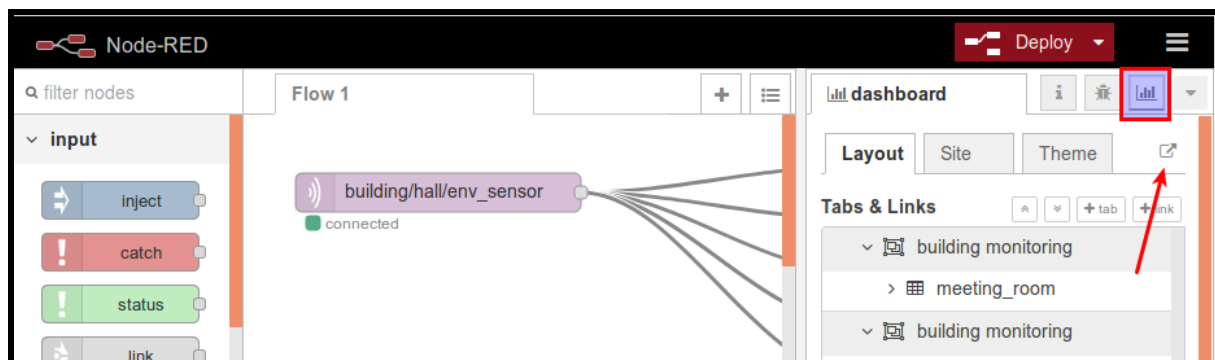
- Select Theme

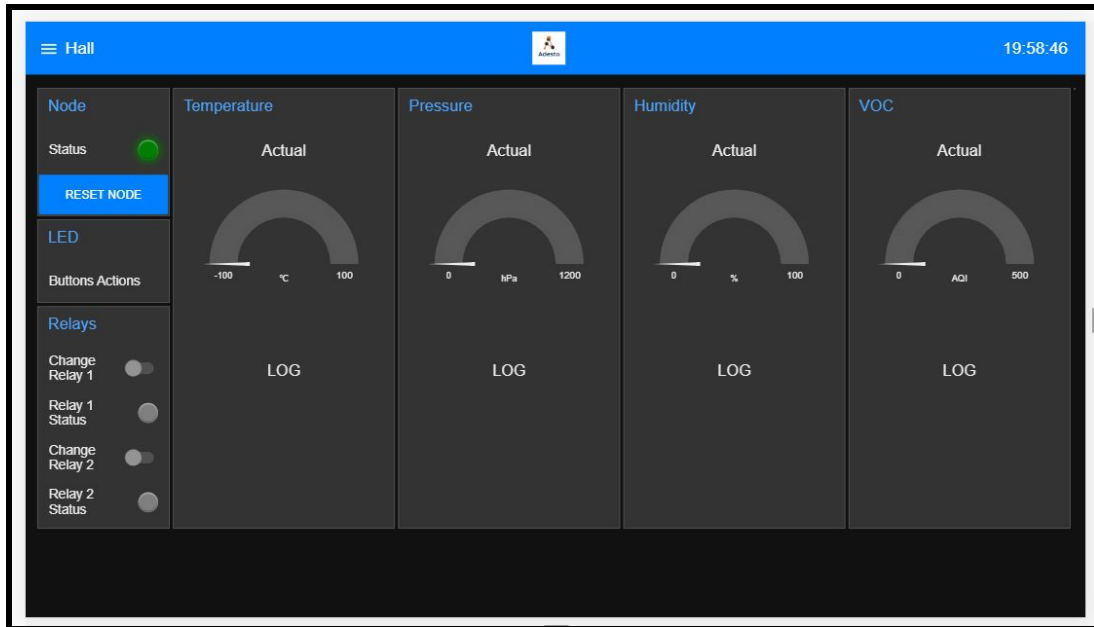


- Deploy flow

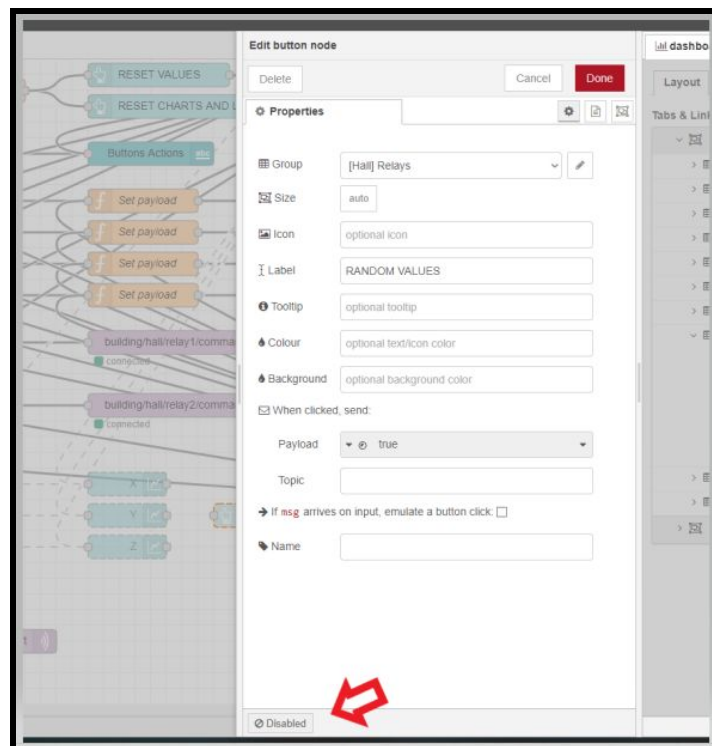


- Start dashboard (is fixed to 1280x720 resolution)





- Try to change data (**ONLY FOR TESTS**)
 - Commands on RPi
 - `mosquitto_pub -t 'building/hall/temperature' -m '{"temperature":24}'`
 - Random button
 - Enable button to display it on the dashboard and click it to put random values



- Dashboard with data



Run simulated data generator

The `sensor_data_publisher.py` is located in `usrlibs/pi/sensor_data_publisher.py`
 The program will generate mqtt messages with simulated data.

- Install prerequisites:
 - `sudo apt install python3 python3-pip`
 - `pip3 install --user pyyaml paho-mqtt docopt`
- Run the utility:
 - `cd usrlibs/pi/`
 - `python3 sensor_data_publisher.py`

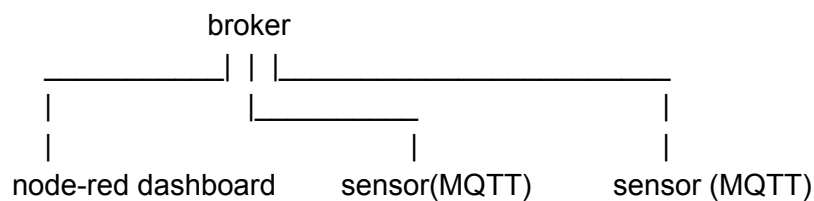
Set-up and run ftmq_mqtt_bridge

The `ftmq_mqtt` bridge is located in [usrlibs/Raspberry_Pi/ftmq_mqtt_bridge](#)
 The program will translate messages from mqtt to ftmq. It needs the MQTT broker running and the ftmq serial device connected.

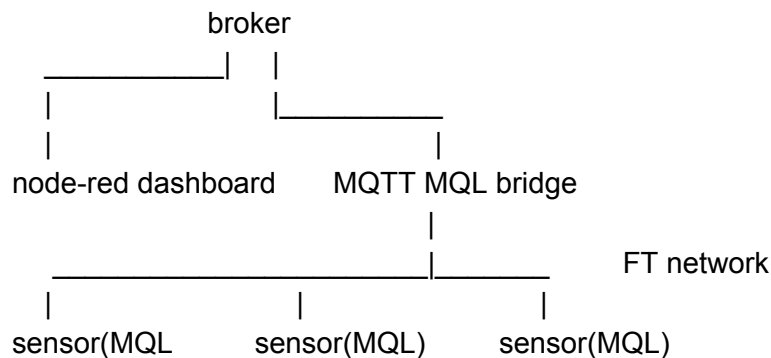
- Install prerequisites:
 - `sudo apt install python3 python3-pip`
 - `pip3 install --user pyserial pyyaml paho-mqtt docopt`
- Copy the `ftmq_bridge` directory to `/etc/ftmq_bridge`
 - `sudo mkdir /etc/ftmq_bridge`

- sudo cp usr/libs/pi/ftmq_bridge/* /etc/ftmq_bridge
- sudo chmod +x /etc/ftmq_bridge/run_ftmq_bridge.sh
- Copy the ftmq_bridge service, enable and start it
 - sudo cp usr/libs/pi/ftmq_bridge.service /lib/systemd/system/
 - Sudo systemctl enable ftmq_bridge.service
 - sudo systemctl start ftmq_bridge.service
 - sudo systemctl status ftmq_bridge.service

MQTT demo



MQL plugged in



an MQL node connects to the MQTT MQL bridge
 subscribes to everything on MQL bus, and republish to the MQTT broker
 subscriptions received in the MQL bus have to be submitted also to the MQTT broker.
 MQTT broker sends messages matching subscriptions, and the bridge has to publish them on the MQL bus.

Connect FT Wiring

Connect the FT twisted pair wiring to the FT click board.

Test

1. Adjust the RGB LED colour on the Arduino using the colour picker.
2. Adjust the white LED colour using the sliders.
3. Check that the gauges display sensor data from the STM32 Nucleo.

Resources

- Usrlibs repository - GIT repository: [link](#)

Contact

Connectex: info@connect-ex.com