ETHERNET WEATHER STATION

The objective is to build a little weather station (temperature, atmospheric pressure). The data will be searchable via the Ethernet port. (web page Management). In addition, you will learn about NUCLEO STMicroelectronics cards while using the I²C protocol and Ethernet frame analysis.

1 NUCLEO STM32 cards	• Discover NUCLEO boards. • IDE ARDUINO installation.
2 NUCLEO programming	• Get started withNUCLEO F429ZI card • Leds and keys board control
3 MEMS et I²C sensors	• MEMS X-NUCLEO-IKS01A2 structural analysis • LPS22HB temperature ans pressure sensor implementation • I²C analysis
4 Ethernet and WEB server	• Static IP and DHCP • Wheather datas access(TELNET) and Wireshark analysis • WEB server

Work requested & answers to be completed on this document by inserting texts and images (screenshots) under the different questions.

1 NUCLEO STM32 BOARD

Discover NUCLEO boards.

Q1-1 From the document flstm32nucleo.pdf, what are the 3 ranges of NUCLEO cards?

http://www.st.com/resource/en/flyer/flstm32nucleo.pdf

Q1-2 What is the role of flash memory in a microcontroller?

https://fr.wikipedia.org/wiki/M%C3%A9moire_flash

Q1-3 What are the main differences between these three NUCLEO ranges?

Q1-4 The microcontroller used is a STM32F429ZI. Complete the information below

http://www.st.com/content/st_com/en/products/microcontrollers/stm32-32-bit-arm-cortex-mcus/stm32-high-performance-mcus/stm32f4-series/stm32f429-439.html?querycriteria=productId=LN1806

Caractéristiques	
Operating Frequency (MHz)	
FLASH Size (kB)	
Internal RAM Size (kB)	
Timers (16 bit)	
Timers (32 bit)	
A/D Converters (12bits)	
D/A Converters (12 bit)	
I/O	
i2C	
SPI	
USART	
UART	
Additional Interfaces	
Supply max Voltage (V)	

IDE ARDUINO Installation

The whole procedure is indicated HERE https://github.com/stm32duino/wiki/wiki

Start by installing the Arduino IDE: https://www.arduino.cc/en/Main/Software

Finish CORE STM32 installation.

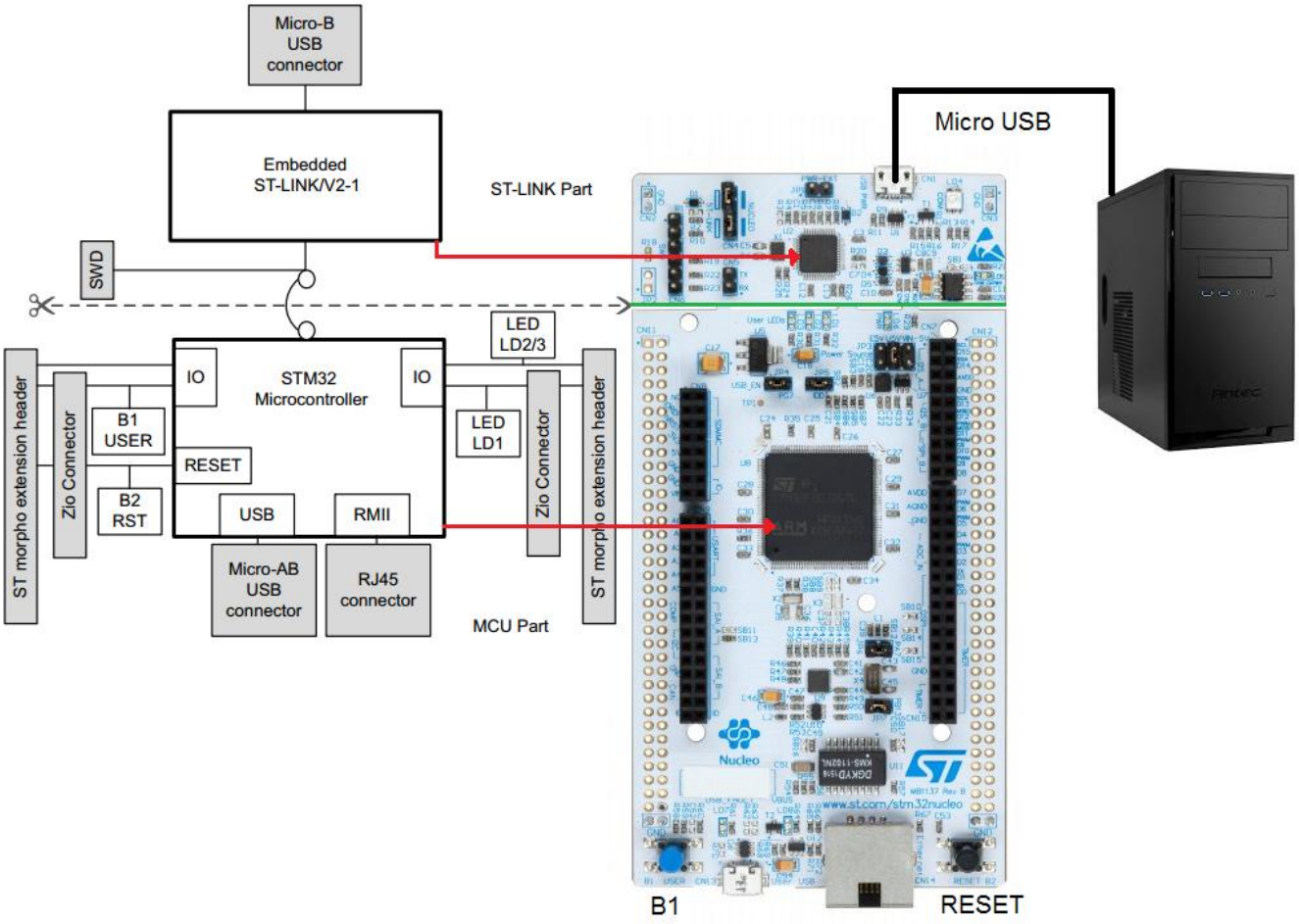
https://github.com/stm32duino/wiki/wiki/Getting-Started

https://github.com/stm32duino/wiki/wiki/Boards-Manager

The list of supported NUCLEO cards is listed here:
https://github.com/stm32duino/Arduino_Core_STM32/blob/master/README.md#boards-available

2 NUCLEO programming

Get started with the NUCLEO F429ZI card



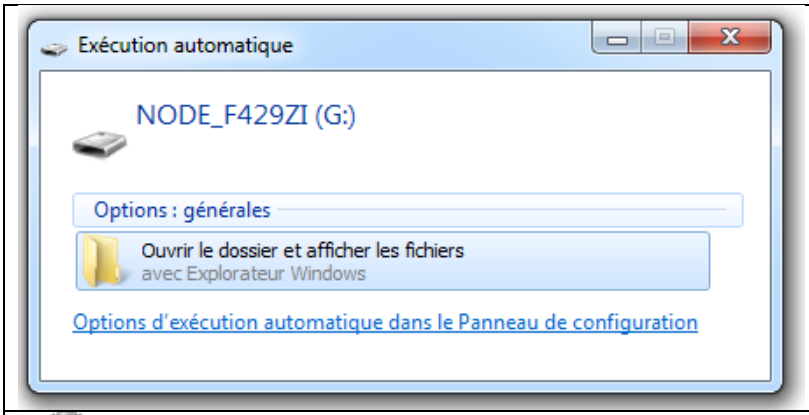
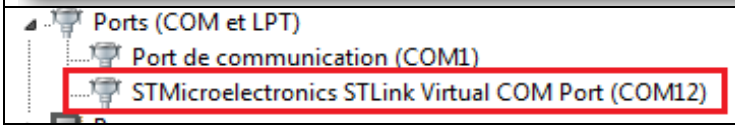
The NUCLEO F429ZI card has a breakable part. This is a programming interface called ST-LINK. It is included on all cards in the NUCLEO range, so there is no need to use a separate programmer.

Note on the nucleo cards drivers

When connecting a NUCLEO card to a PC, two drivers are needed

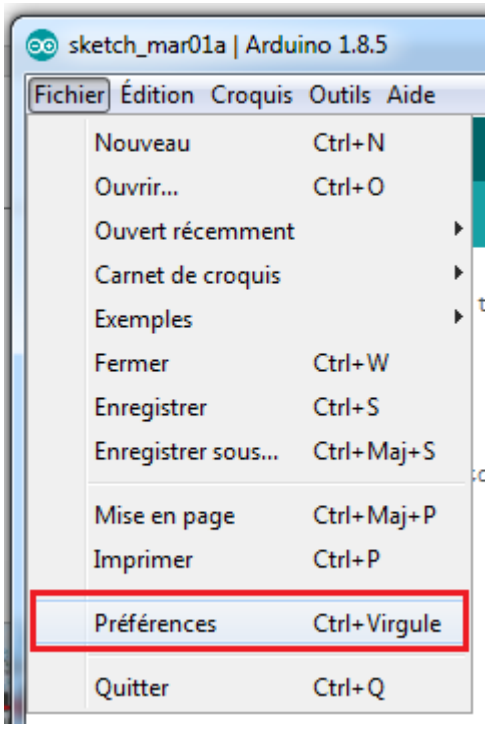
One is installed automatically, it is a USB mass Storage Device as for a USB key.

The other driver is the ST-LINK driver. In this case it will be necessary to install the driver manually.

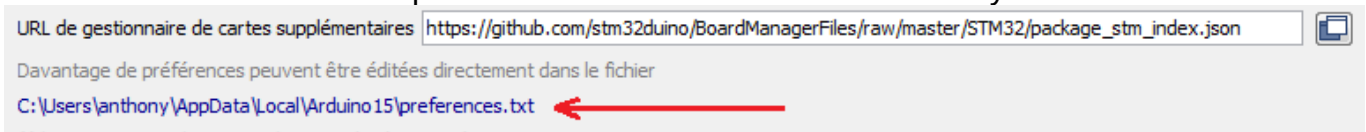
	USB mass Storage Device. Automatic driver installation
	Manual driver installation from the Device Manager.

Where is the ST-LINK driver ?

In the Arduino IDE, go to the "Preferences" menu



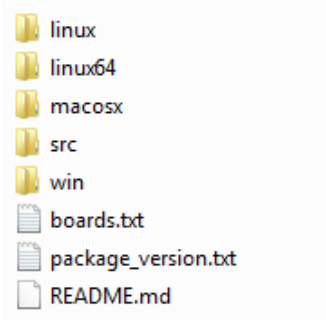
Click the link above. That's opens additional cards installation directory



Continue to browse directories and access tools

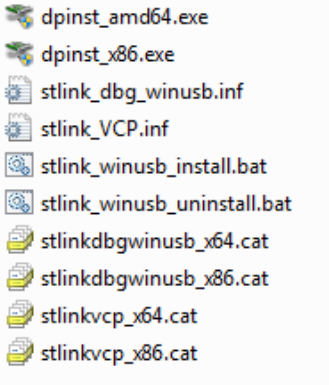
C:\Users\name\AppData\Local\Arduino15**\packages\STM32\tools\STM32Tools\2017.9.22\tools**

At this level, a choice is required depending on your operating system



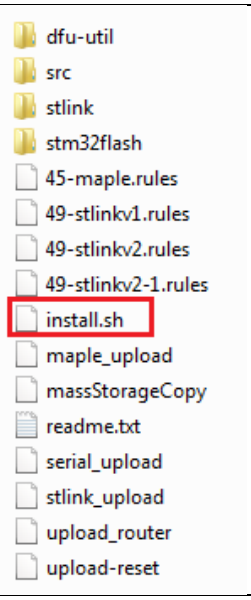
For Windows:

C:\Users\name\AppData\Local\Arduino15\packages\STM32\tools**\STM32Tools\2017.9.22\tools\win\stlink\ST-LINK_USB_V2_Driver**

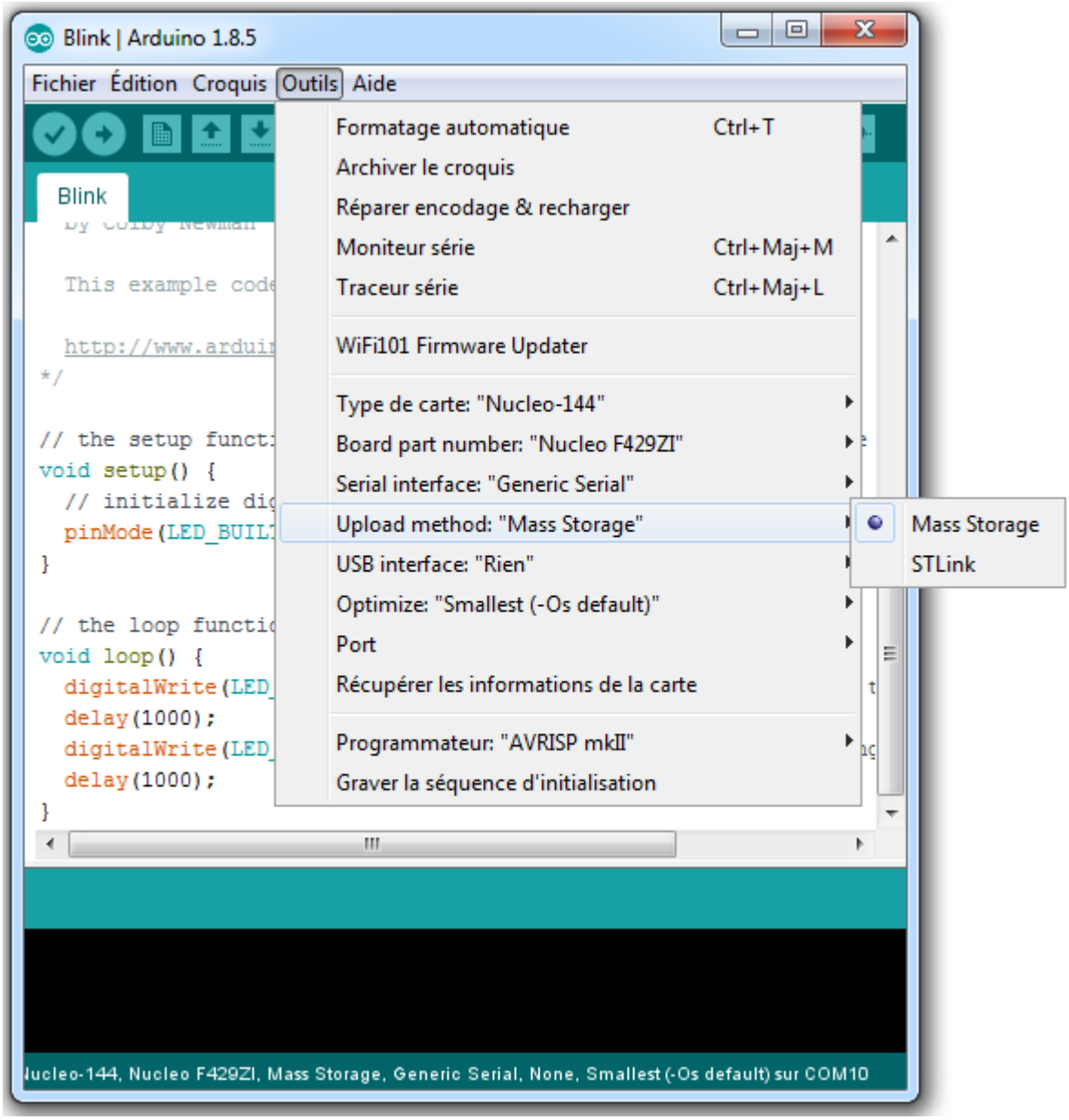
	In Device Manager, perform a manual installation by pointing this directory.

For Linux:

/home/name/.arduino15/packages/STM32/tools**/STM32Tools/2017.9.22/tools/linux64**

	Run the install.sh script as root sudo bash install.sh

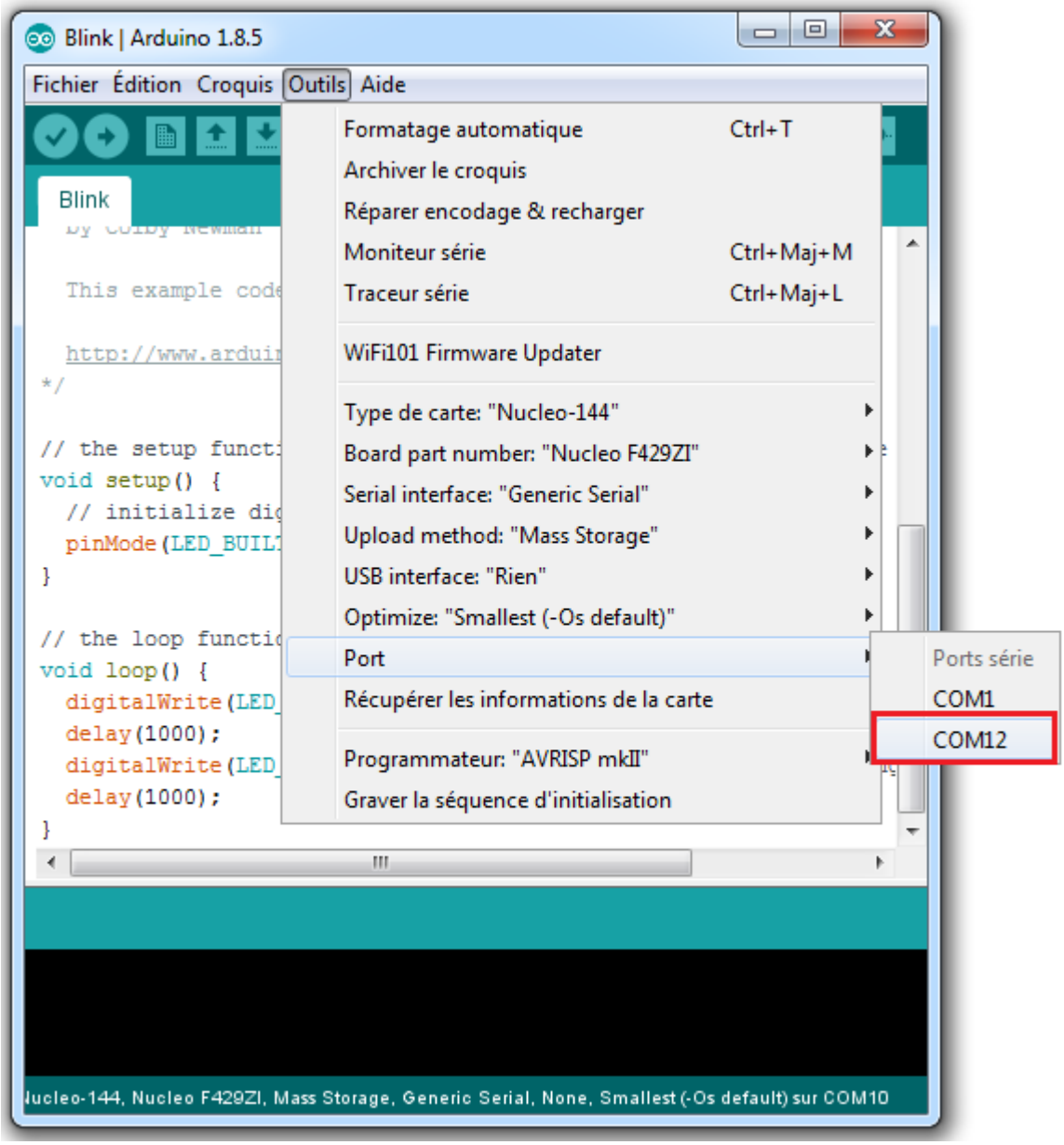
In the Arduino environment, configure the Tool menu as follows:



There are two modes of programming Mass Storage or STLink

Choose **Mass Storage** by default

Check that a communication port is installed correctly. (Other than COM1)



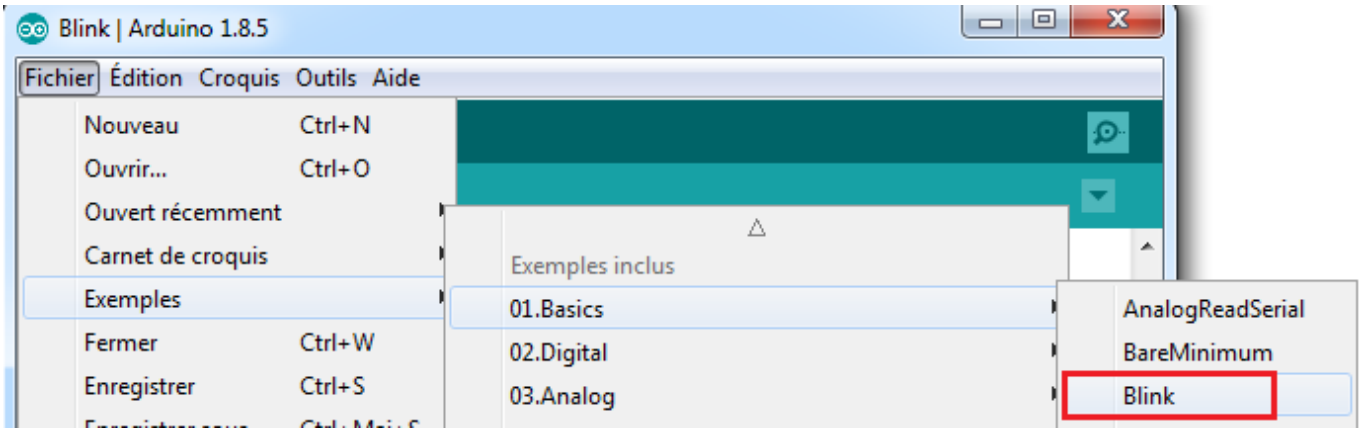
Under linux the communication port is named **ttyACM0**
Dans la console taper :

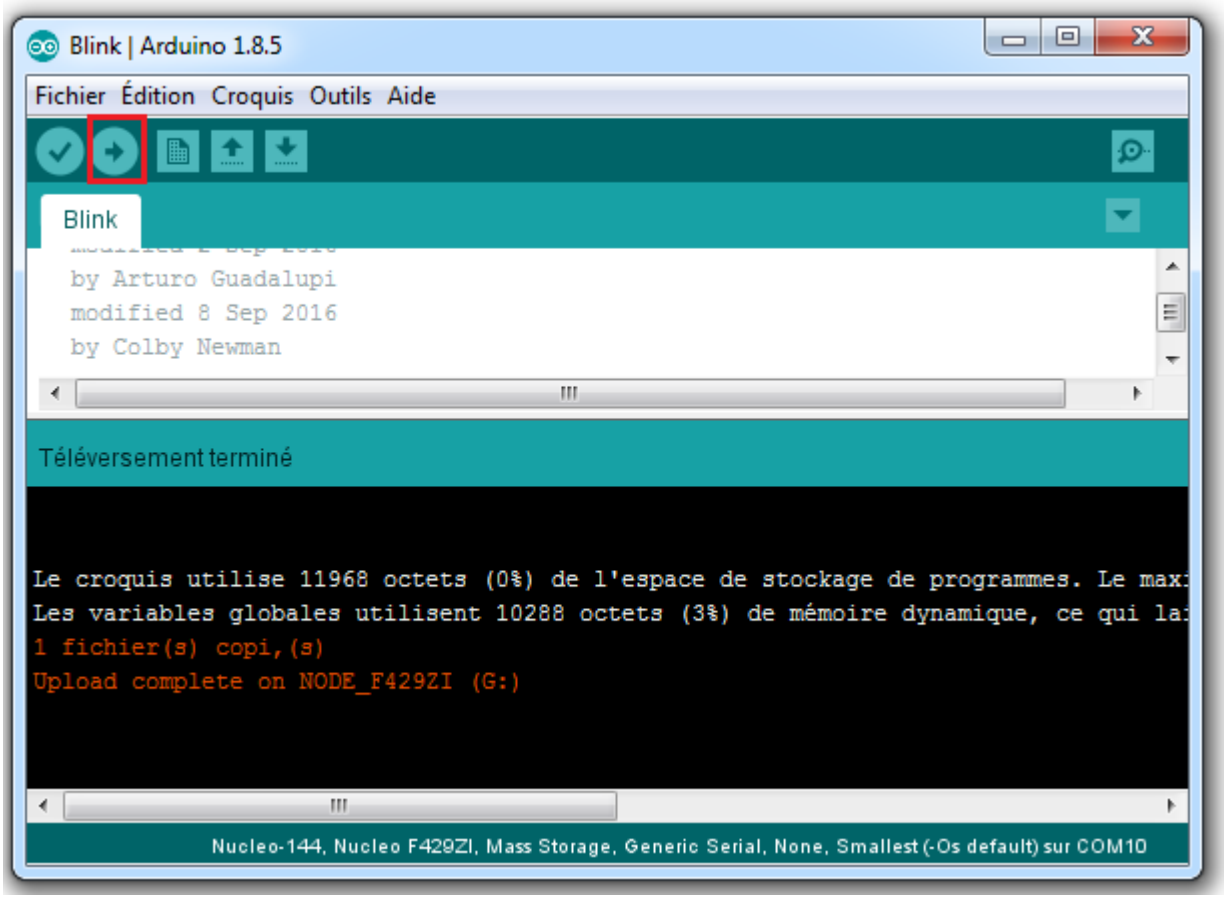
ls /dev/ttyA*

/dev/ttyACM0 should appear.

LED and push button management

Load the "Blink" program from the examples



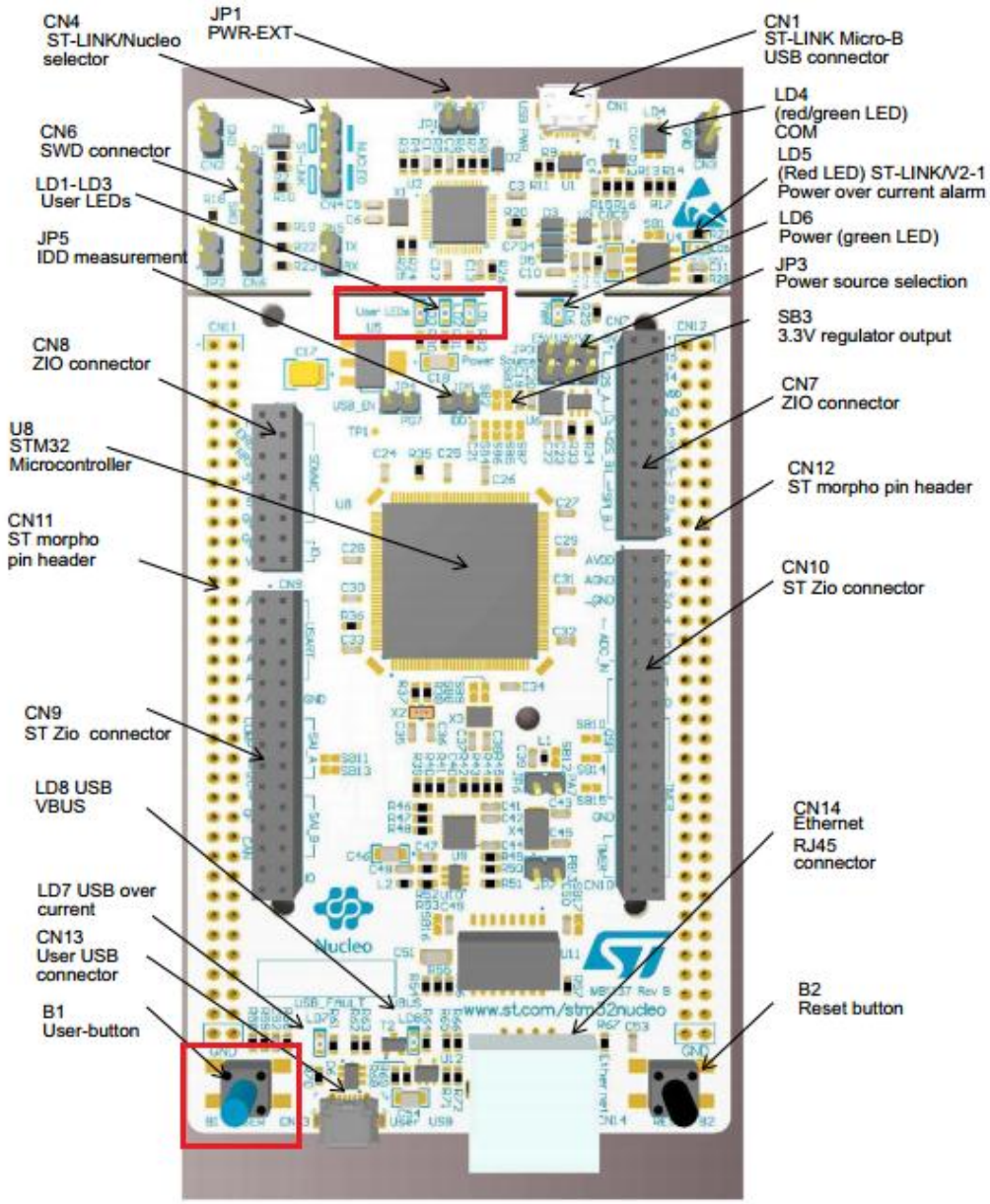


Led LD1 should flash every second.

Replace the LED_BUILTIN constant with PB0.

Once the program is uploaded. There must be no change.

Locate the LD1, LD2, LD3 LEDs and the User pushbutton on the board



Q2-1 Complete the following table:

Pxx : STM32 NUCLEO 144 input / output port, from the document en.DM00244518.pdf, pages 30-31 or in annex 1
http://www.st.com/resource/en/user_manual/dm00244518.pdf

Active level: low or high

	LD1	LD2	LD3	BP USER
Pxx				
Active level				
Led color				

Q2-2 Find the 3 Leds and the push button in the structural diagram of the NUCLEO board

http://www.st.com/resource/en/schematic_pack/nucleo_144pins_sch.zip

Q2-3 Complete the following program to scroll through the 3 Leds

```
/*
scrolling leds
*/
#define LD1 PB0
#define LD2
#define LD3

void setup() {

  pinMode(LD1, OUTPUT);
  pinMode(   ,   );
  pinMode(   ,   );
}

void loop() {
  digitalWrite(LD1, HIGH);
  delay(100);




}
```

Q2-4 What does the following program do?

```
/*
  Bp user test
*/
#define LD1 PB0
#define BPUSER PC13

byte bpUserState;

void setup() {
  pinMode(LD1, OUTPUT);
  pinMode(BPUSER, INPUT);
}

void loop() {
  bpUserState = digitalRead(BPUSER);
  if (bpUserState == 1) {
    digitalWrite(LD1, HIGH);
  }
  else
  {
    digitalWrite(LD1, LOW);
  }
```

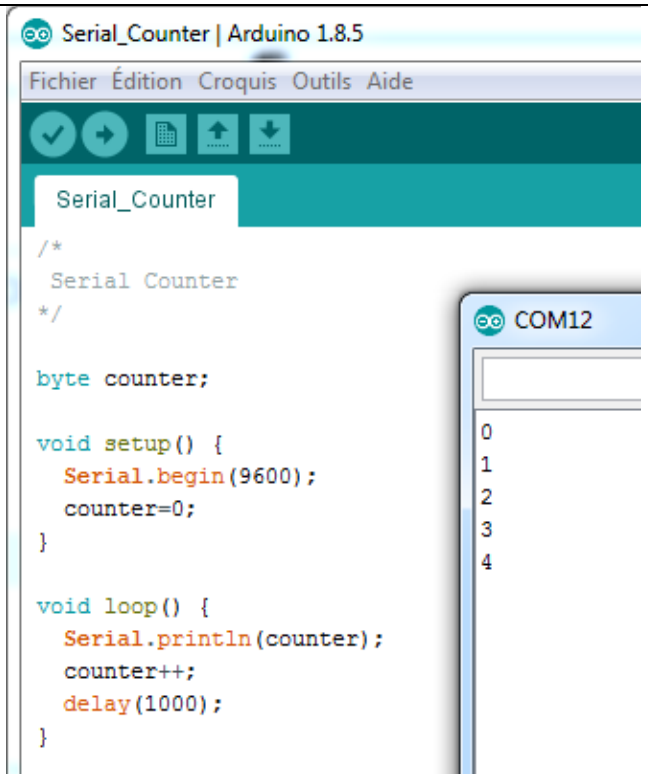
Q2-5 Modify the previous program in order to scroll LD1, LD2 and LD3 LEDs in the green, blue, red direction when the USER pushbutton is released then in the red, blue, green direction when the USER push button is pressed.

Q2-6 What does the following program do?

Do not forget to open the serial monitor 

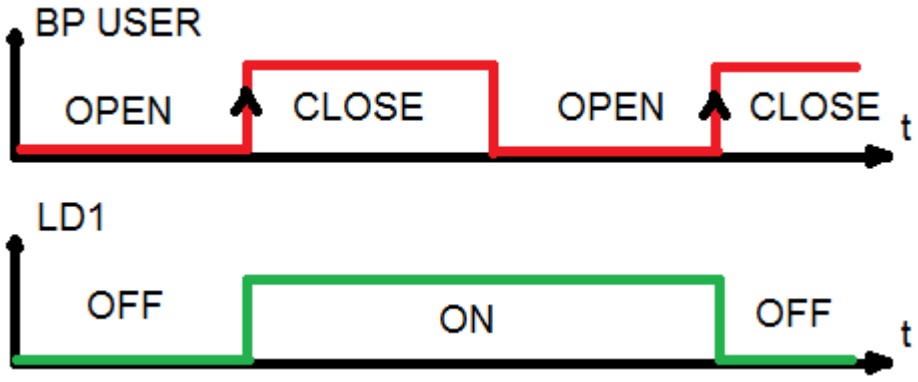
on the right port (COMx ou ttyACM0)

```
/*
 Serial Counter
*/

int counter;

void setup() {
  Serial.begin(9600);
  counter=0;
}

void loop() {
  Serial.println(counter);
  counter++;
  delay(1000);
}
```



Q2-7 Modify the previous program to increment the variable when pressing the USER pushbutton. Then display this variable on the serial monitor. Each time the push button is pressed, the LED LD1 changes state.

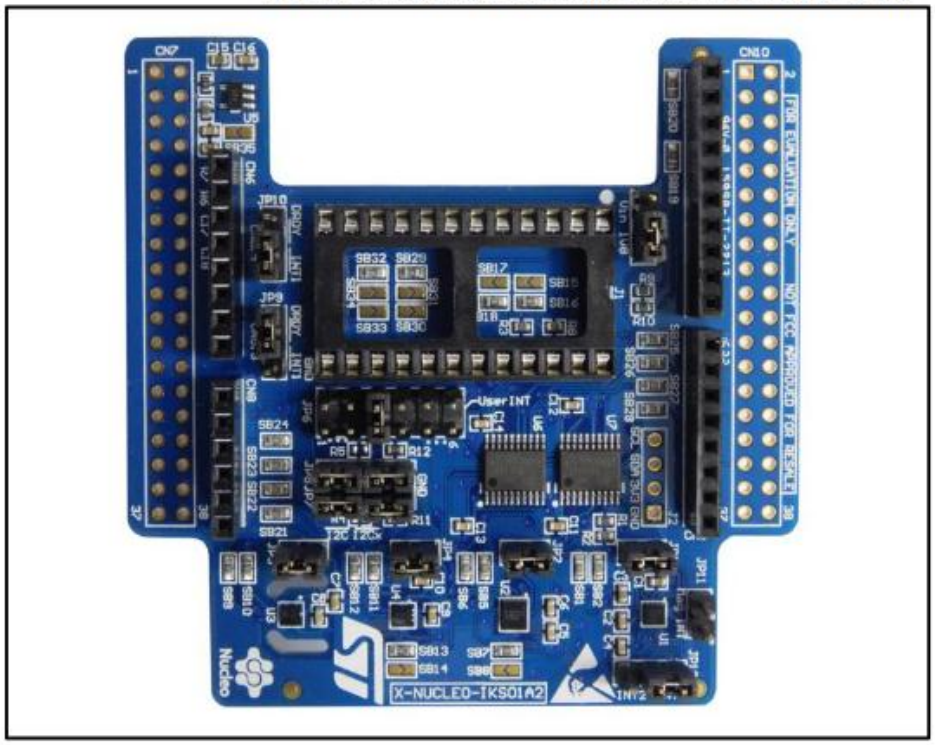
Beware, it is necessary to detect a rising edge (low level to high level)



3 MEMS et I²C Senor

x-nucleo-iks01a2 Shield board

X-NUCLEO-IKS01A2 expansion board (sheild)



Q3-1 Define a **MEMS** using the following links

https://players.brightcove.net/618591172001/rydXtRRN_default/index.html?videoId=2289006872001

http://www.st.com/en/mems-and-sensors.html

https://fr.wikipedia.org/wiki/Microsyst%C3%A8me_%C3%A9lectrom%C3%A9canique

MEMS X-NUCLEO-IKS01A2 structural analysis

Q3-2 From dm00333132.pdf (page1), what are the 4 sensors used on the Sheild map? Specify their functions.

http://www.st.com/resource/en/user_manual/dm00333132.pdf

- L
- L
- H
- L

Q3-3 From the document x-nucleo-iks01a2_schematic.pdf, associate the reference of the 4 sensors with U1, U2, U3 and U4. Locate the sensors on the schematic.

http://www.st.com/resource/en/schematic_pack/x-nucleo-iks01a2_schematic.pdf

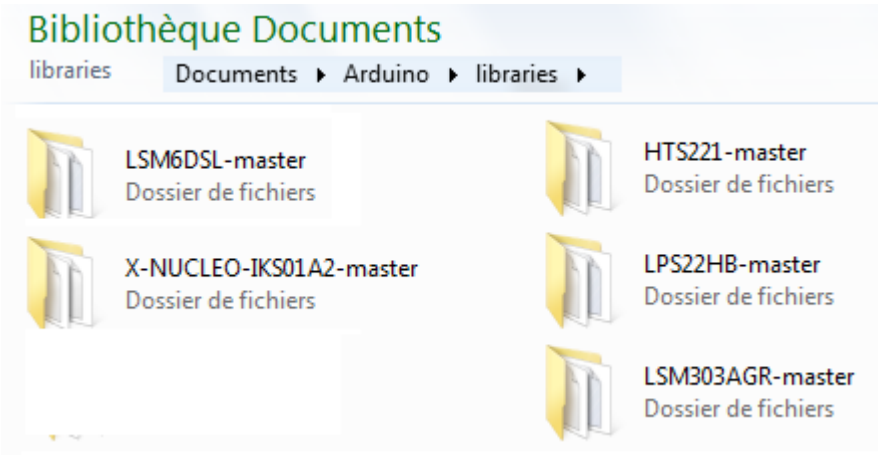
U1	U2	U3	U4

Q3-4 What is the role of jumper JP1, JP2, JP3 and JP4? justify.

Libraries Installation
Ci-dessous les liens vers les librairies ainsi que les documentations STM des 4 capteurs.

https://github.com/stm32duino/LSM6DSL	www.st.com/resource/en/datasheet/lsm6dsl.pdf
https://github.com/stm32duino/LSM303AGR	www.st.com/resource/en/datasheet/lsm303agr.pdf
https://github.com/stm32duino/HTS221	www.st.com/resource/ja/datasheet/hts221.pdf
https://github.com/stm32duino/LPS22HB	www.st.com/resource/en/datasheet/dm00140895.pdf
https://github.com/stm32duino/X-NUCLEO-IKS01A2	

Directory Content C:\Users\name\Documents\Arduino\libraries



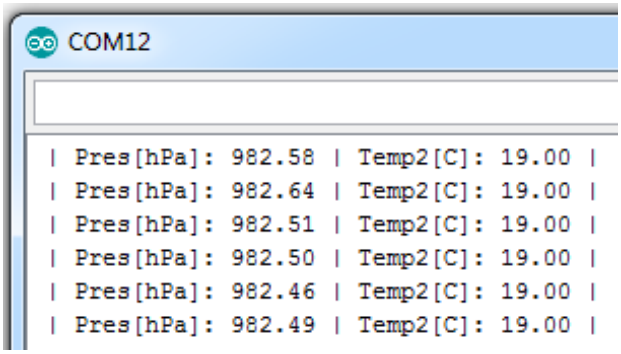
Temperature and pressure sensor LPS22HB Implementation

Q3-5 What does the following program do?

Do not forget to open the serial monitor 

on the right port (COMx ou ttyACM0) and at the right speed.

```
/**
  temp_Pressure_LPS22HB
*/

#include <LPS22HBSensor.h>
#define sync 13

LPS22HBSensor  *PressTemp;

void setup() {
  // Led.
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(sync, OUTPUT);

  // Initialize serial for output.
  Serial.begin(115200);

  // Initialize I2C bus.
  Wire.begin();

  // Initlialize components.

  PressTemp = new LPS22HBSensor (&Wire);
  PressTemp->Enable();
}

void loop() {
  // Led blinking.
  digitalWrite(LED_BUILTIN, HIGH);
  delay(250);
  digitalWrite(LED_BUILTIN, LOW);
  delay(250);


  // Read pressure and temperature.
  float pressure, temperature;
  digitalWrite(sync, HIGH);
  PressTemp->GetPressure(&pressure);
  PressTemp->GetTemperature(&temperature);
  digitalWrite(sync, LOW);

  // Output data.
  Serial.print(" | Pres[hPa]: ");
  Serial.print(pressure, 2);
  Serial.print(" | Temp[C]: ");
  Serial.print(temperature, 2);
  Serial.println(" |");

}
```



This program ….

I²C comunication analysis

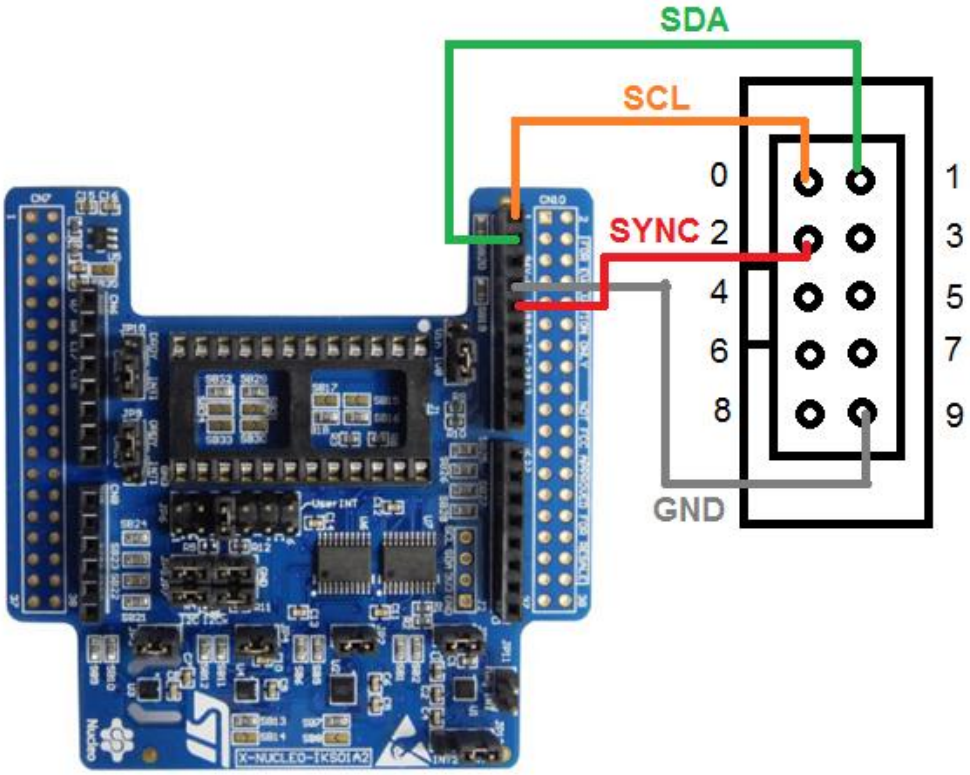
In order to understand the I²C bus communication, it is necessary to obtain a logic analyzer.

The Saleae analyzer is available on ebay

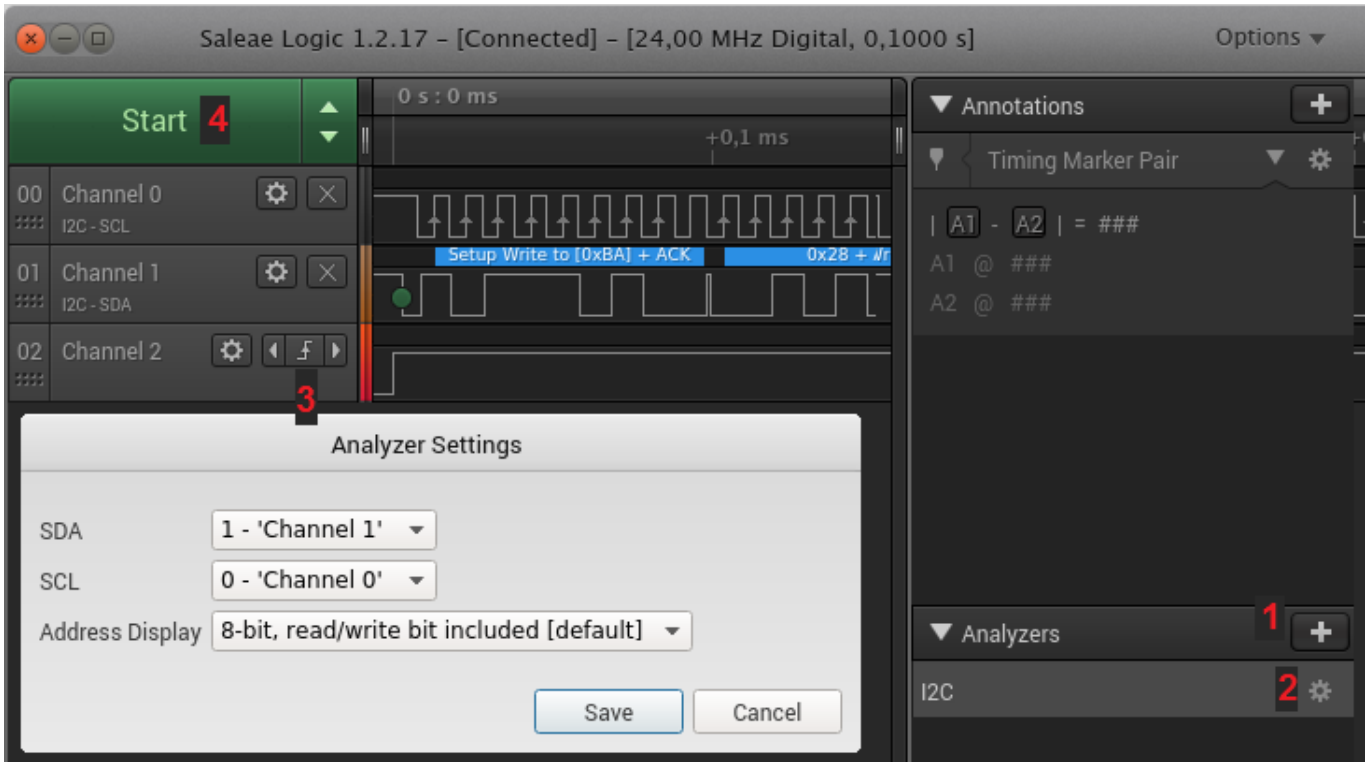
https://www.ebay.fr/itm/24MHz-8CH-USB-Logic-Analyzer-8-Channel-Logic-Analyzer-Compatible-to-Saleae/172316908968

The software is downloadable at the following address: https://www.saleae.com/fr/downloads

The I²C bus is on the CN5 sheild card connector. **Connect** the logic analyzer as follows:



Configure the Saleae software as shown in the screenshot below:



1- Add the I²C analysis protocol
2- Configure the I²C bus on channel 0 for SCL, 1 for SDA.
3- Set a rising edge on channel 2
4- Start an acquisition (Start)

Q3-6 What is the logical level of the U4 CS input, cavalier SB13 (LPS22HB) ?

http://www.st.com/resource/en/schematic_pack/x-nucleo-iks01a2_schematic.pdf

Q3-7 Quel est le rôle de la broche CS ? (Page 9 hts221.pdf)

www.st.com/resource/ja/datasheet/hts221.pdf

Q3-8 What is the role of pin SD0 / SA0? (page 9 hts221.pdf) Give the logical level on the schématics

www.st.com/resource/ja/datasheet/hts221.pdf

Q3-9 Then give the two hexadecimal values of the read and write command on the I²C bus.? (Page 26 hts221.pdf)

www.st.com/resource/ja/datasheet/hts221.pdf

Table 11. SAD+Read/Write patterns

Command	SAD[6:1]	SAD[0] = SA0	R/W	SAD+R/W
Read	101110	0	1	10111001 (B9h)
Write	101110	0	0	10111000 (B8h)
Read	101110	1	1	10111011 (BBh)
Write	101110	1	0	10111010 (BAh)

Q3-10 Deduce the 7-bit i²C address of the component in hexadecimal.

Q3-11 The following questions will be about the hts221.pdf constructor document (page 32) related to the following screenshots if the logic analyzer is not used. Otherwise, analyze the received frames.

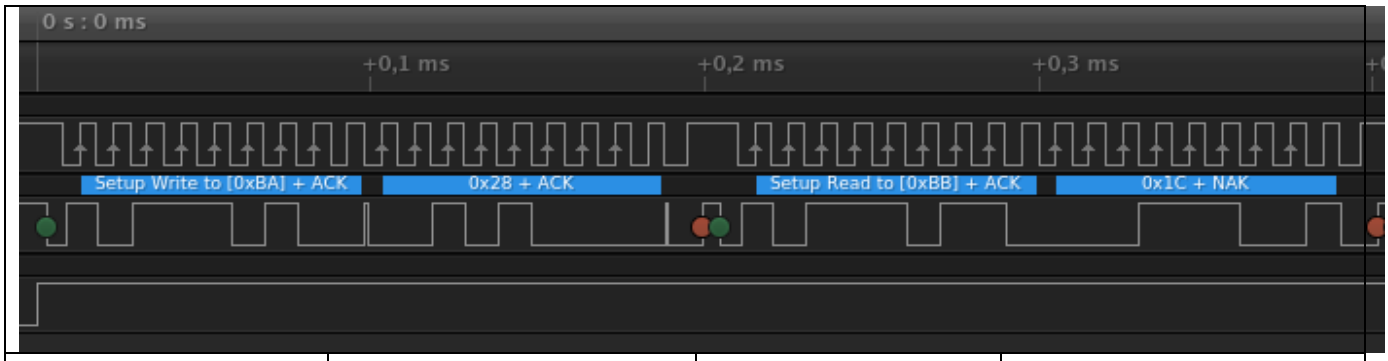
www.st.com/resource/ja/datasheet/hts221.pdf

The hts221 sensor library uses a 4-byte frame to read the contents of a one-byte register. The frame breaks down as follows:

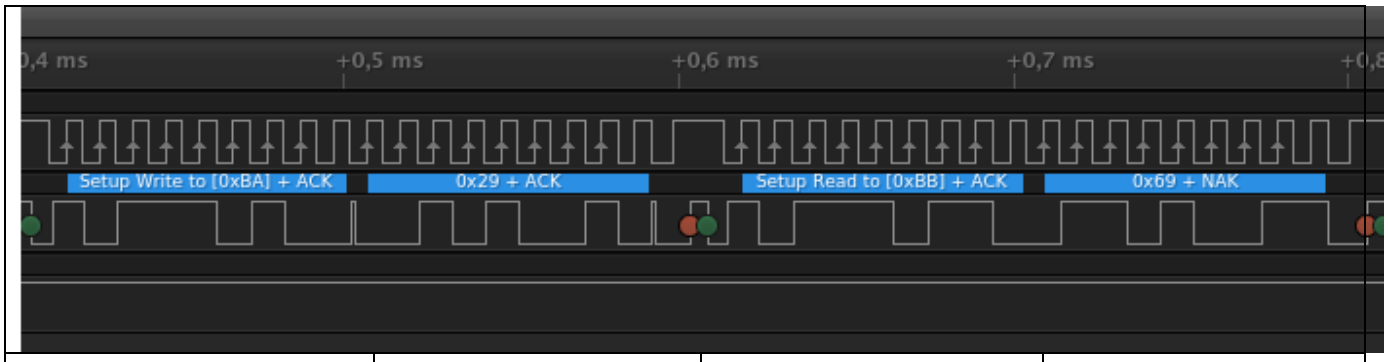
I²C write operation	Register selection	I²C reading operation	Reading the register

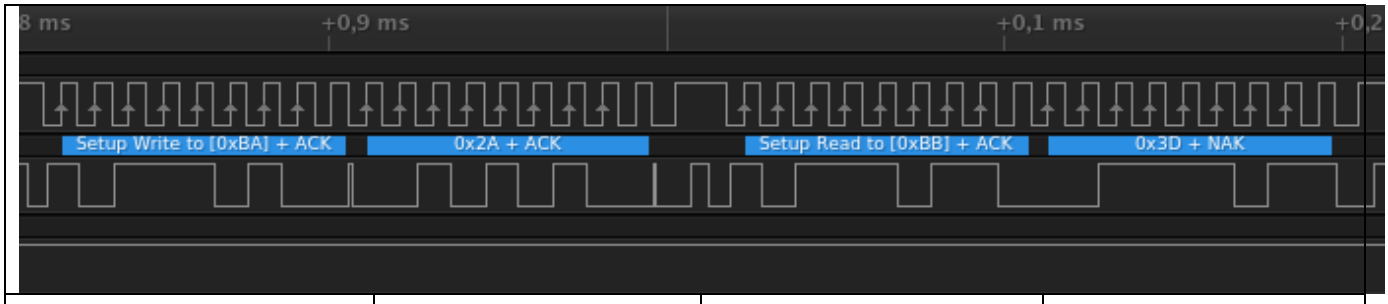
Complete the following acquisition tables as shown in the first example:

Pressure reading

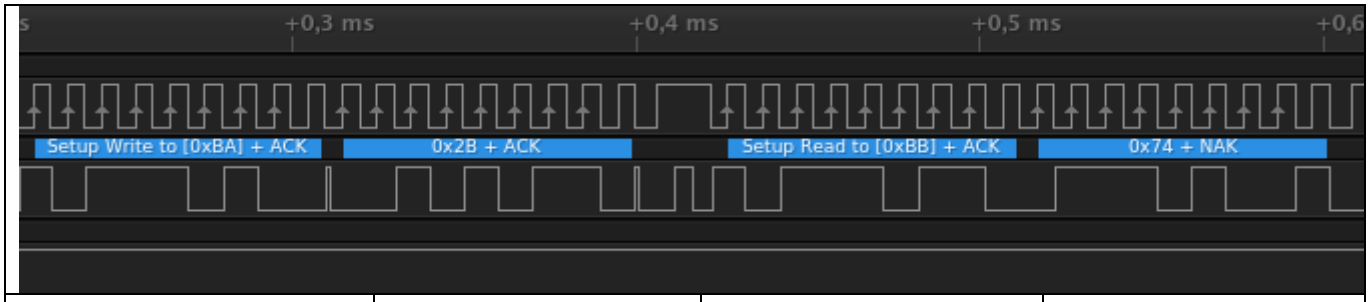


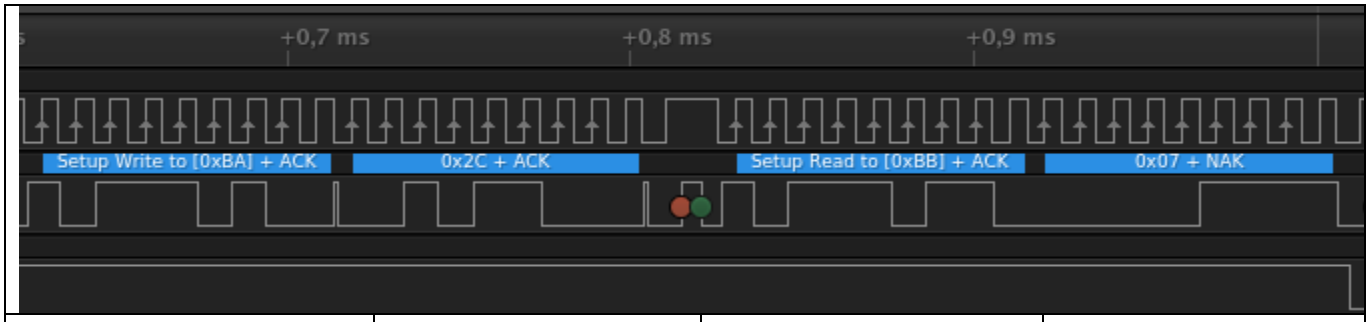
I²C Write opération 0xBA	Registrer selection **PRESS_OUT_XL** 0x28	I²C Reading operation 0xBB	**Register reading PRESS_OUT_XL** Valeur 0x1c





Temperature reading





Results analysis

Q3-12 Complete the following table with regard to the pressure sensor

Registers	**PRESS_OUT_H**	**PRESS_OUT_L**	**PRESS_OUT_XL**
Hexadécimal 8 bits			
Décimal (24 bits)			

Q3-13 Deduce the pressure in hectopascal hts221.pdf (page 16) What is the measuring range of the sensor?
www.st.com/resource/ja/datasheet/hts221.pdf

Q3-14 Compléter le tableau suivant pour le capteur de température

Registers	**TEMP_OUT_H**	**TEMP_OUT_L**
Hexadécimal 8 bits		
Décimal (16 bits)		

Q3-15 What is the variable type for the previous decimal value? (strike out the mention of useless) Why? hts221.pdf (page 46)

unsigned int	**int**

Q3-16 The temperature is calculated by dividing by 10 the 16-bit decimal value. Calculate the temperature

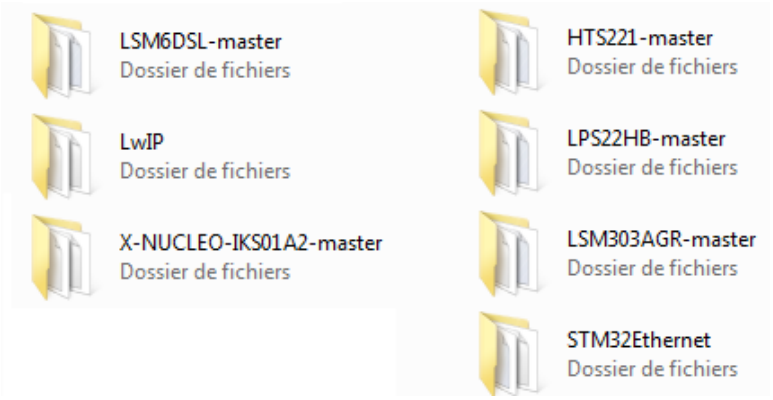
4 Ethernet and WEB server

libraries Installation

Below links to libraries

https://github.com/stm32duino/STM32Ethernet
https://github.com/stm32duino/LwIP

Directory Content C:\Users\name\Documents\Arduino\libraries



The STM32 Ethernet library is equivalent to Arduino Ethernet and described here:

https://www.arduino.cc/en/Reference/Ethernet

DHCP addressing and IP address display

Q4-1 From the following page, complete the program to display the address mask, gateway, and DNS.

Use the following methods:

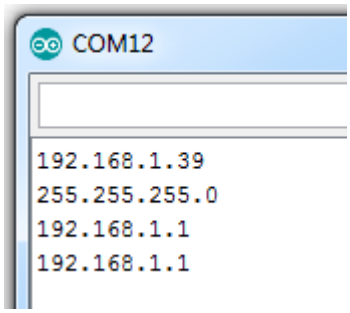
Ethernet.subnetMask()

Ethernet.gatewayIP()

Ethernet.dnsServerIP()

```
/*
  DHCP-based IP printer
*/

#include <LwIP.h>
#include <STM32Ethernet.h>

// Enter a MAC address for your controller below.
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  // start the Ethernet connection:

  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    while (1) {}
  }
  // print your local IP address:
  Serial.println(Ethernet.localIP());
  Serial.println(      );
  Serial.println(      );
  Serial.println(      );

}

void loop() {

}
```

To see the IP address displayed in the serial monitor, make a RESET of the card once the program has been uploaded. (The serial monitor must be open beforehand)



Static IP address

Q4-2 See the link below. Modify the previous program to perform static IP address.

https://www.arduino.cc/en/Reference/EthernetIPAddress

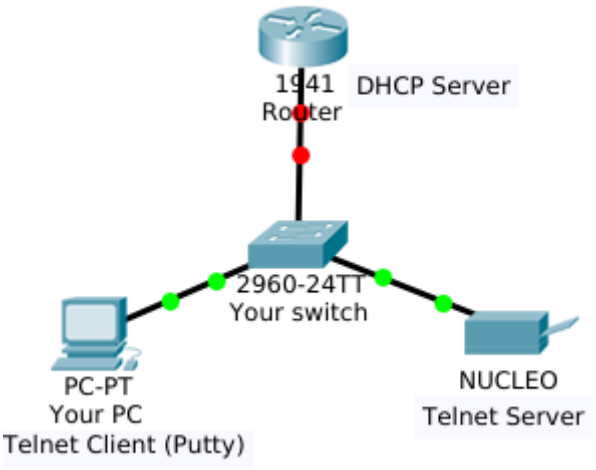
```
/*
  Fixed IP printer
 */

#include <LwIP.h>
#include <STM32Ethernet.h>

// Enter a MAC address for your controller below.
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};

IPAddress ip(     );

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  // start the Ethernet connection:

  Ethernet.begin(     );

  // print your local IP address:
  Serial.println(Ethernet.localIP());

}

void loop() {

}
```

Check the connection with a ping command from console

ping 192.168.1.177

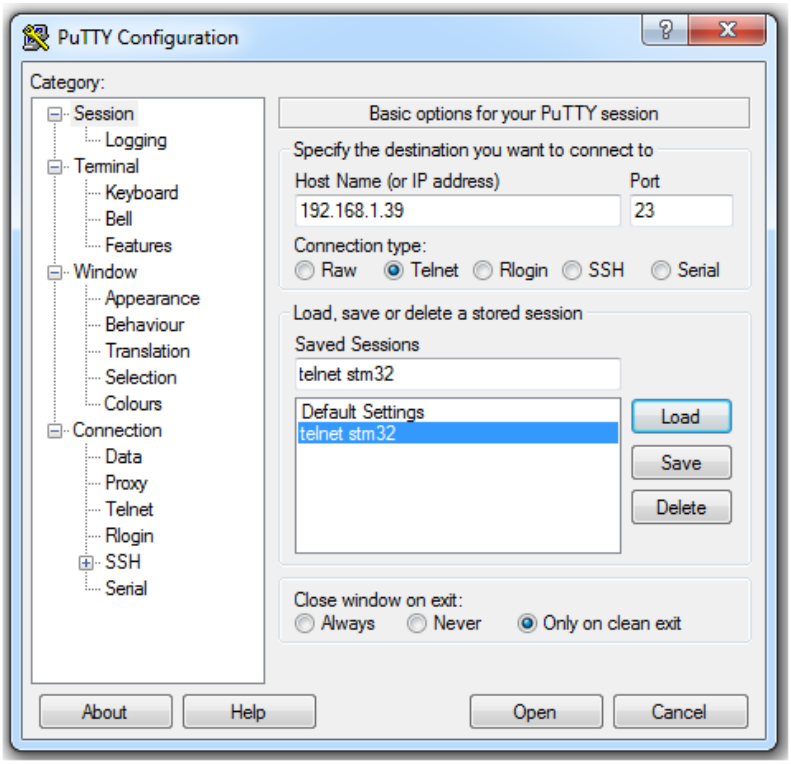
```
Envoi d'une requête 'Ping'  192.168.1.177 avec 32 octets
Réponse de 192.168.1.177 : octets=32 temps=1 ms TTL=255
Réponse de 192.168.1.177 : octets=32 temps<1ms TTL=255
Réponse de 192.168.1.177 : octets=32 temps<1ms TTL=255
Réponse de 192.168.1.177 : octets=32 temps<1ms TTL=255
```

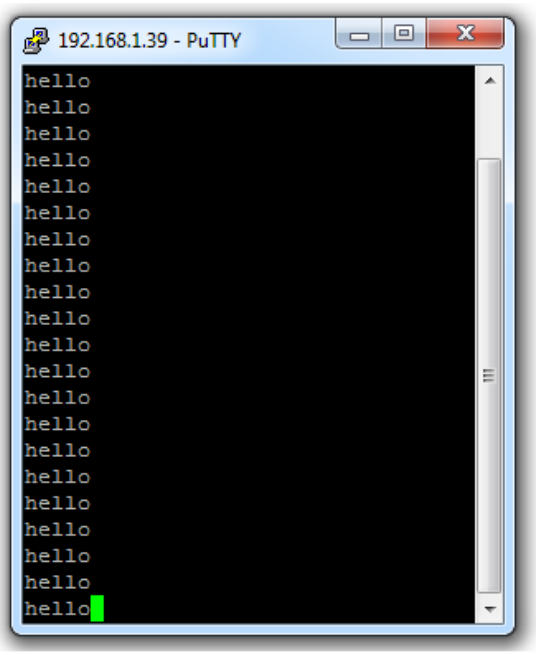
Telnet access



Upload the program on the next page, Check with putty the communication with the server

https://www.putty.org/

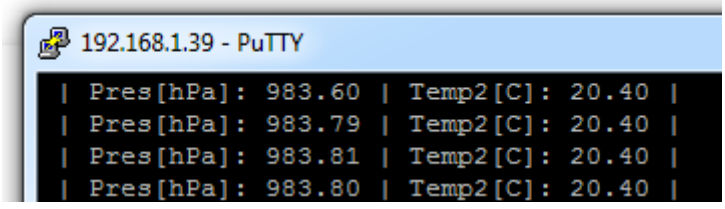




Warning Telnet server only accepts one client at a time

```
/*
 telnet Server
 */

#include <LwIP.h>
#include <STM32Ethernet.h>

// Enter a MAC address for your controller below.
byte mac[] = {0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED};

// telnet defaults to port 23
EthernetServer server(23);
EthernetClient client;

boolean alreadyConnected = false; // whether or not the client was connected previously

void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  // start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
    while (1) {}
  }
  // start listening for clients
  server.begin();

  Serial.print("Telnet server address: ");
  Serial.println(Ethernet.localIP());
}

void loop() {
  // wait for a new client
   client = server.available();

  // when the client connected, say hello every second:
  if (client) {
      server.println("hello");
      delay(1000);
  }
}
```

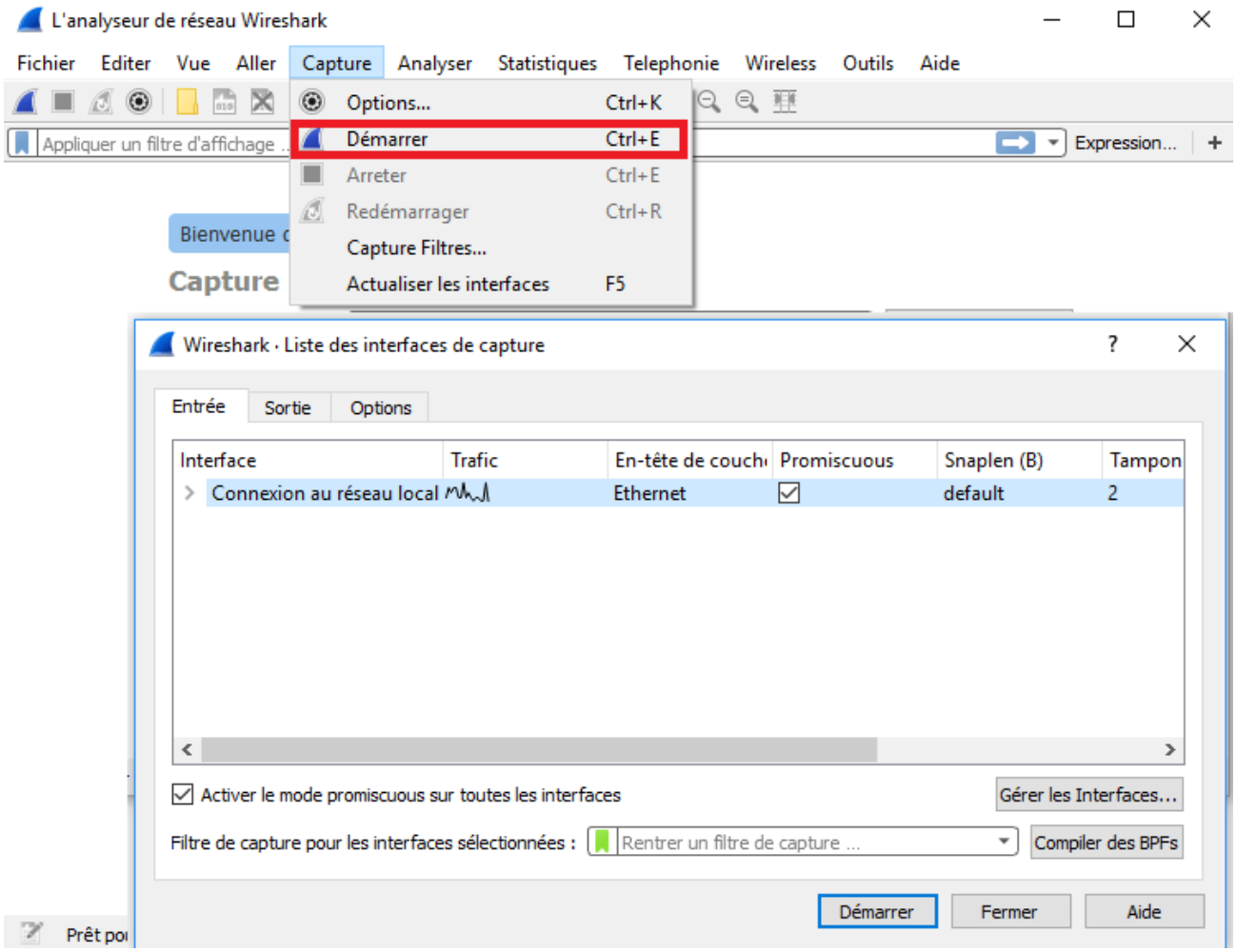
Q4-3 Using question **Q3-5** and the previous program, display the hts221 sensor data with TELNET

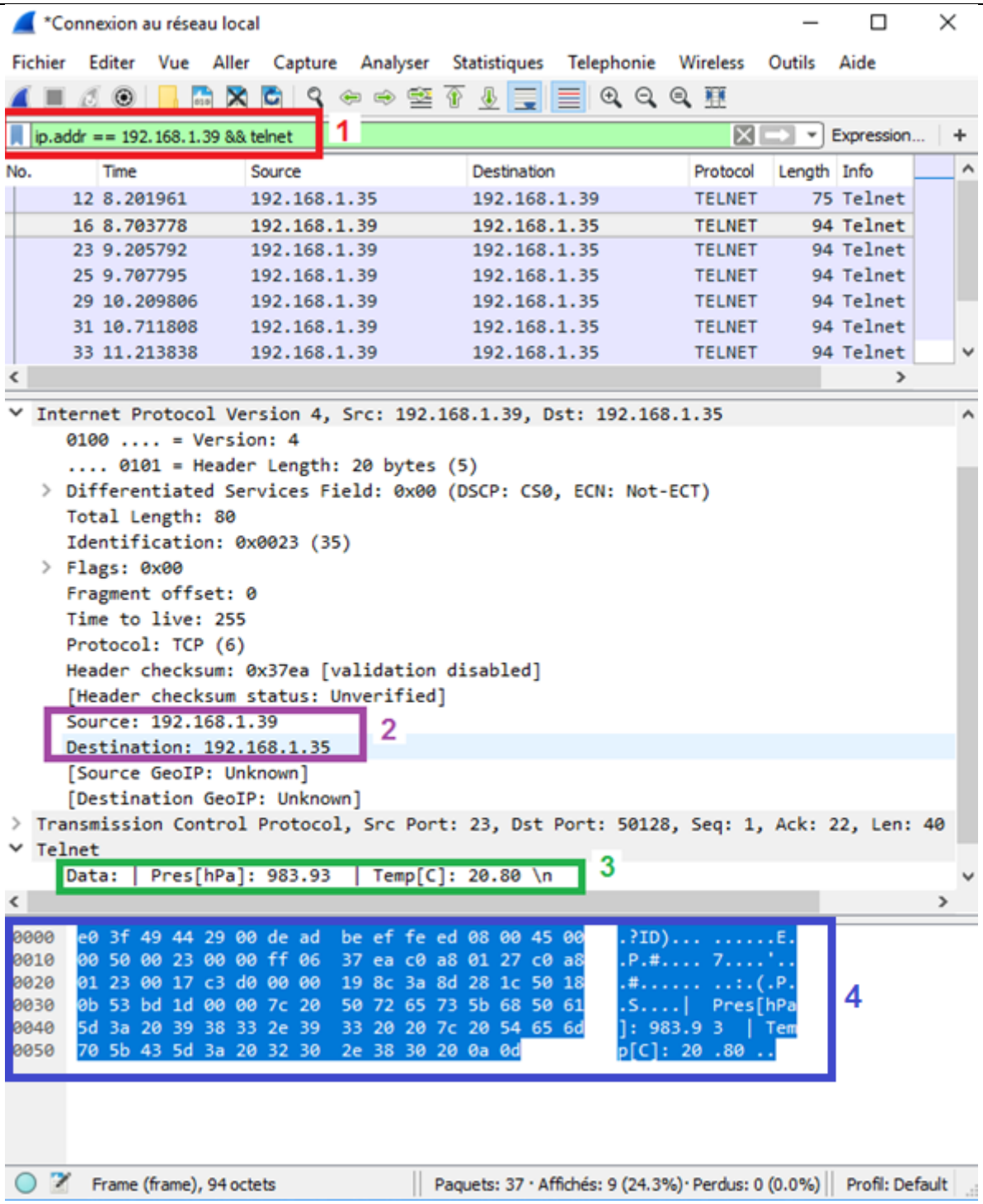


WireShark analysis

https://www.wireshark.org/

Install and run the WireShark program





1- Use a filter on server IP analysis and telnet,
2- Source and destination IP addresses,
3- data
4- Pack of data

Q4-4 Is the data sent encrypted?

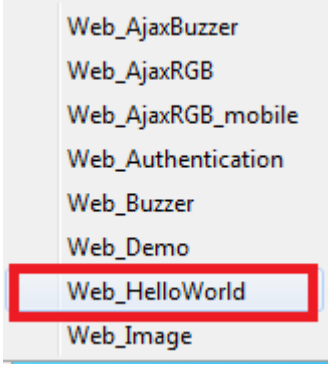
WEB Server

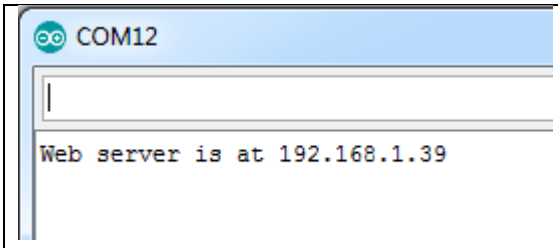
In order to facilitate the management of a WEB server in the NUCLEO card, it is necessary to add the following library:

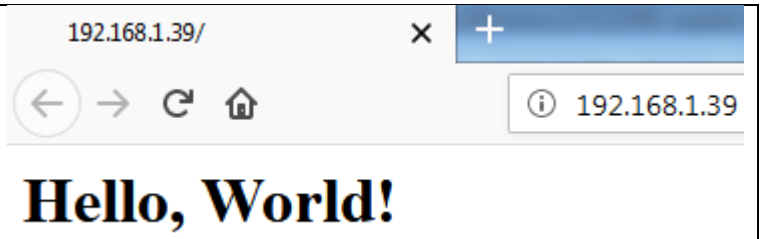
https://github.com/f4goh/Webduino

Or in zip file joined

Start by loading the program **Web_HelloWorld.ino**







Do not forget to use a static IP address

Set up your ip address
IPAddress ip(192, 168, 1, 177);

Then use this method
Ethernet.begin(mac, ip);

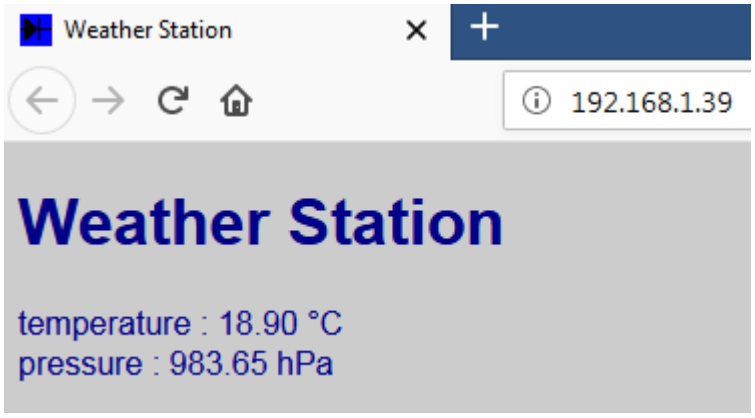
It is possible to specify the gateway, DNS and mask as **Q4.1** question

Q4-5 Implement the following web page instead of HelloWorld

It will be necessary to add the character \ at the end of each line and just before the "of the code HTML

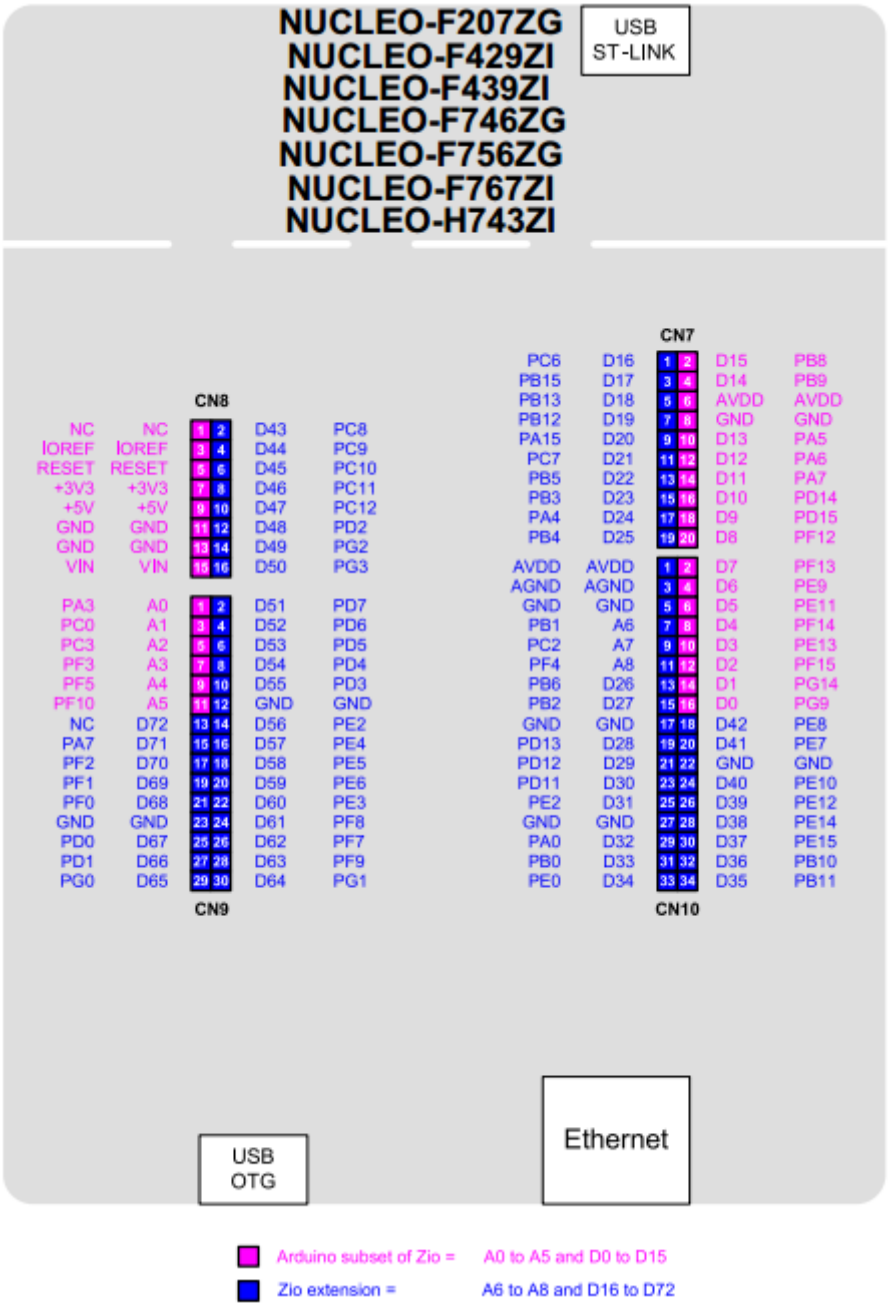
```
<!DOCTYPE html>
<html>
  <head>
    <title> Weather Station</title>
          <meta charset="UTF-8" />
          <meta name="viewport" content="width=device-width, initial-scale=1.0"/>
          <meta http-equiv='refresh' content='5'/>
    <style>
      body { background-color: #cccccc; font-family: Arial, Helvetica, Sans-Serif; Color:
#000088; }
    </style>
  </head>
  <body>
    <h1> Weather Station</h1>
  </body>
</html>
```

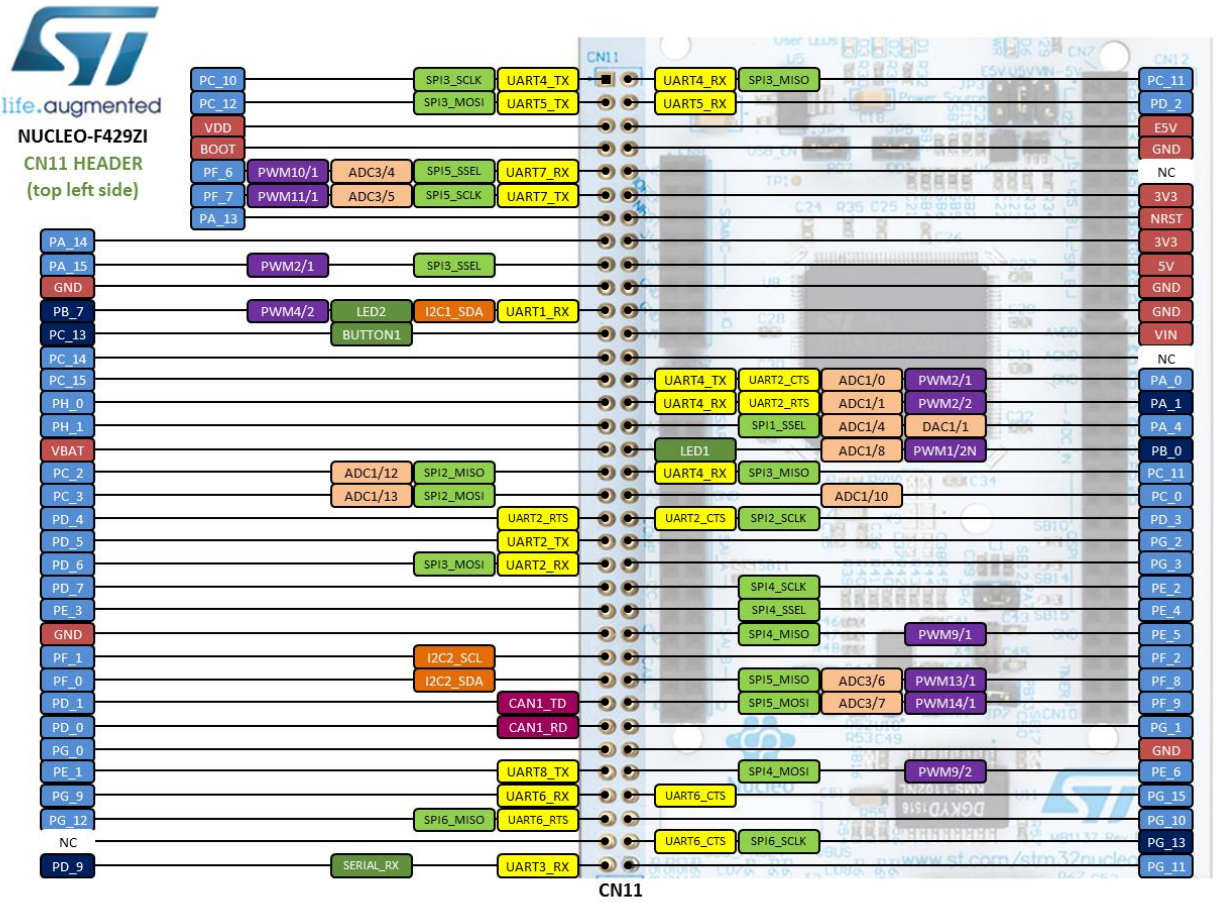
Q4-6 Finally, add the pressure and temperature information to the previous web page



Annex 1

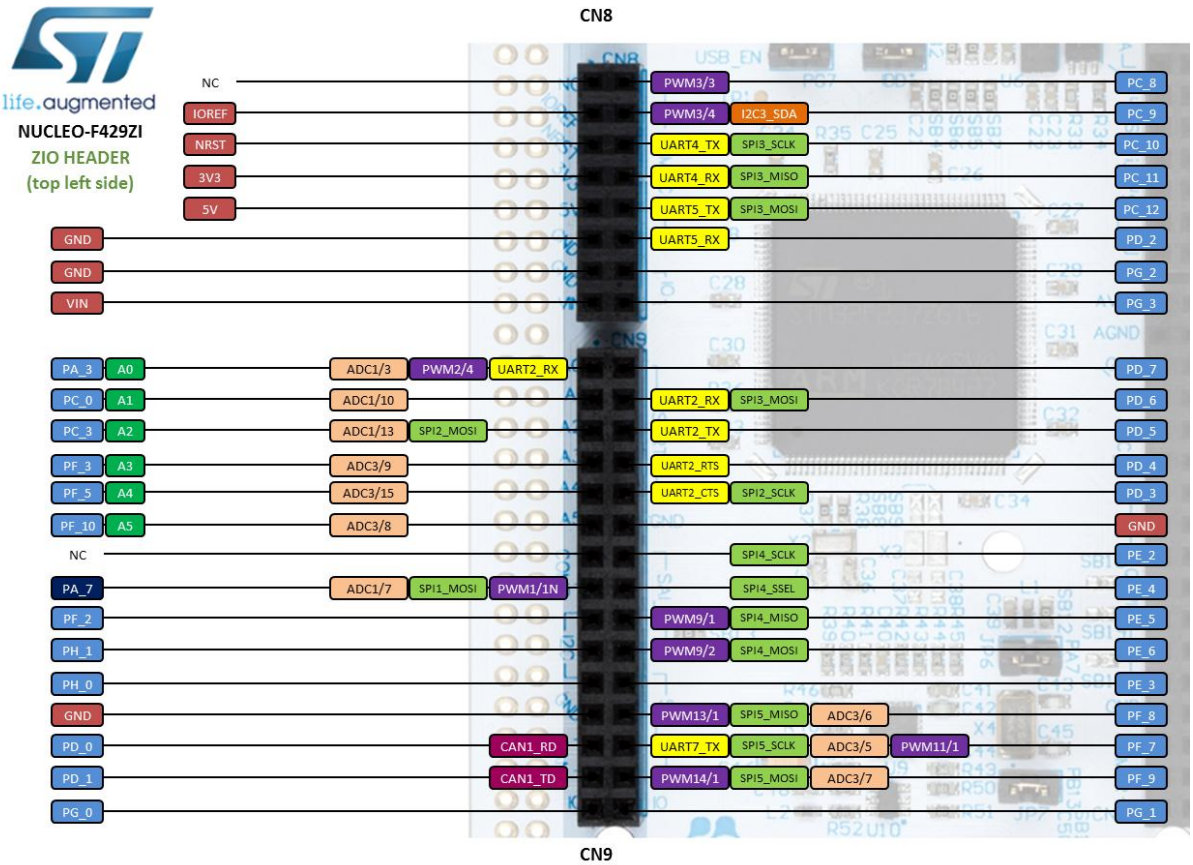
STM32 (Pxx) card input-output port with Arduino equivalences (Dxx)



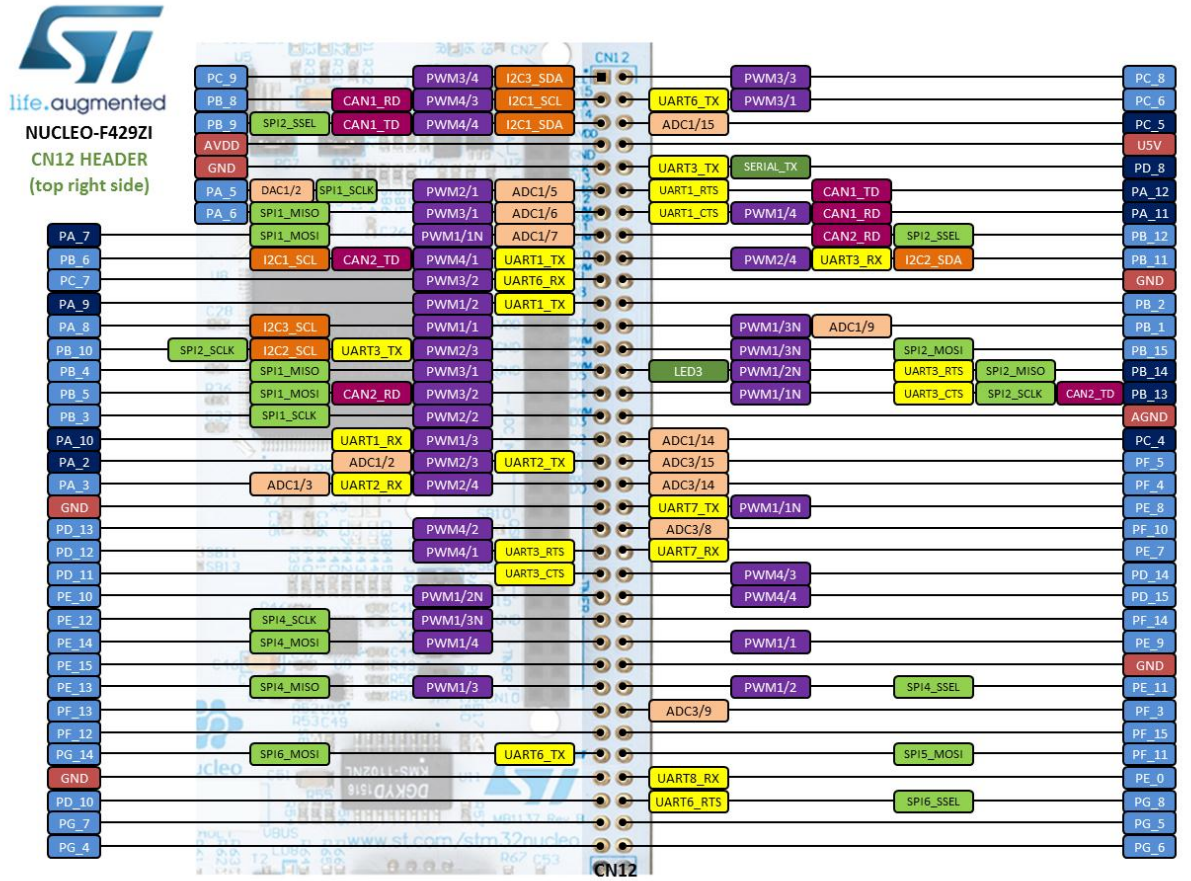

life.augmented
NUCLEO-F429ZI
CN11 HEADER
(top left side)
CN11
PC_10
PC_12
VDD
BOOT
PF_6
PF_7
PA_13
PA_14
PA_15
GND
PB_7
PC_13
PC_14
PC_15
PH_0
PH_1
VBAT
PC_2
PC_3
PD_4
PD_5
PD_6
PD_7
PE_3
GND
PF_1
PF_0
PD_1
PD_0
PG_0
PE_1
PG_9
PG_12
NC
PD_9
SPI3_SCLK
SPI3_MOSI
PWM10/1
PWM11/1
ADC3/4
ADC3/5
SPI5_SSEL
SPI5_SCLK
UART7_RX
UART7_TX
PWM2/1
SPI3_SSEL
PWM4/2
LED2
BUTTON1
I2C1_SDA
UART1_RX
ADC1/12
ADC1/13
SPI2_MISO
SPI2_MOSI
UART2_RTS
UART2_TX
UART2_RX
SPI3_MOSI
I2C2_SCL
I2C2_SDA
CAN1_TD
CAN1_RD
UART8_TX
UART6_RX
SPI6_MISO
UART6_RTS
SERIAL_RX
UART3_RX
UART4_TX
UART4_RX
UART5_RX
SPI3_MISO
UART2_CTS
UART2_RTS
SPI1_SSEL
LED1
SPI2_SCLK
SPI4_SCLK
SPI4_SSEL
SPI4_MISO
SPI4_MOSI
SPI5_MISO
SPI5_MOSI
UART6_CTS
SPI6_SCLK
ADC1/0
ADC1/1
ADC1/4
ADC1/8
ADC1/10
ADC3/6
ADC3/7
PWM2/1
PWM2/2
DAC1/1
PWM1/2N
PWM9/1
PWM9/2
PWM13/1
PWM14/1
PC_11
PD_2
E5V
GND
NC
3V3
NRST
3V3
5V
GND
GND
VIN
NC
PA_0
PA_1
PA_4
PB_0
PC_11
PC_0
PD_3
PG_2
PG_3
PE_2
PE_4
PE_5
PF_2
PF_8
PF_9
PG_1
GND
PE_6
PG_15
PG_10
PG_13
PG_11

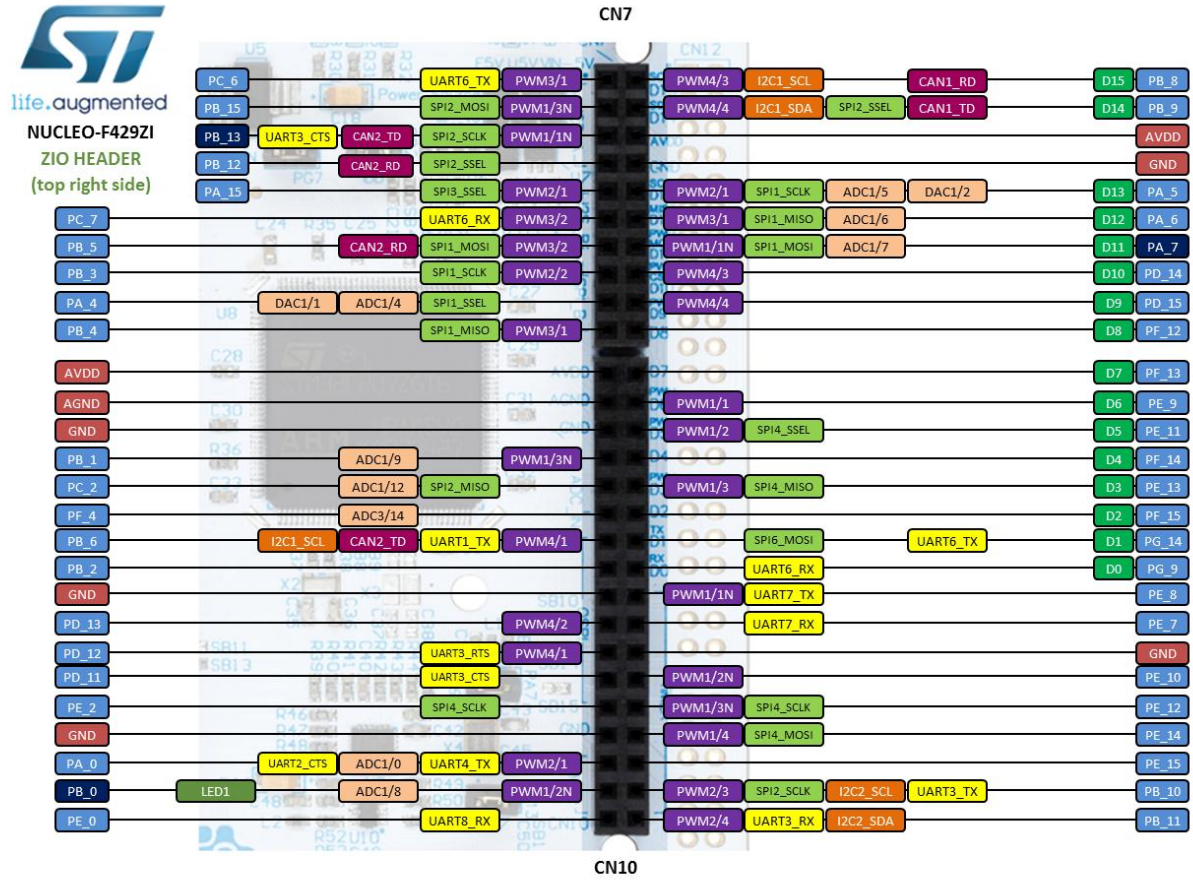
CN11

CN8


life.augmented
NUCLEO-F429ZI
ZIO HEADER
(top left side)
CN8
CN9
NC
IOREF
NRST
3V3
5V
GND
GND
VIN
PA_3
PC_0
PC_3
PF_3
PF_5
PF_10
NC
PA_7
PF_2
PH_1
PH_0
GND
PD_0
PD_1
PG_0
A0
A1
A2
A3
A4
A5
ADC1/3
ADC1/10
ADC1/13
ADC3/9
ADC3/15
ADC3/8
ADC1/7
PWM2/4
UART2_RX
SPI2_MOSI
SPI1_MOSI
PWM1/1N
CAN1_RD
CAN1_TD
PWM3/3
PWM3/4
I2C3_SDA
UART4_TX
UART4_RX
UART5_TX
UART5_RX
SPI3_SCLK
SPI3_MISO
SPI3_MOSI
UART2_TX
UART2_RTS
UART2_CTS
SPI2_SCLK
SPI4_SCLK
SPI4_SSEL
SPI4_MISO
SPI4_MOSI
PWM9/1
PWM9/2
PWM13/1
SPI5_MISO
SPI5_SCLK
SPI5_MOSI
PWM14/1
ADC3/6
ADC3/5
ADC3/7
PWM11/1
PC_8
PC_9
PC_10
PC_11
PC_12
PD_2
PG_2
PG_3
PD_7
PD_6
PD_5
PD_4
PD_3
GND
PE_2
PE_4
PE_5
PE_6
PE_3
PF_8
PF_7
PF_9
PG_1

CN9


life.augmented
NUCLEO-F429ZI
CN12 HEADER
(top right side)
CN12
PC_9 PB_8 PB_9 AVDD GND PA_5 PA_6 PA_7 PB_6 PC_7 PA_9 PA_8 PB_10 PB_4 PB_5 PB_3 PA_10 PA_2 PA_3 GND PD_13 PD_12 PD_11 PE_10 PE_12 PE_14 PE_15 PE_13 PF_13 PF_12 PG_14 GND PD_10 PG_7 PG_4
PWM3/4 PWM4/3 PWM4/4 PWM2/1 PWM3/1 PWM1/1N PWM4/1 PWM3/2 PWM1/2 PWM1/1 PWM2/3 PWM3/1 PWM3/2 PWM2/2 PWM1/3 PWM2/3 PWM2/4 PWM4/2 PWM4/1 PWM1/2N PWM1/3N PWM1/4 PWM1/3
I2C3_SDA I2C1_SCL I2C1_SDA ADC1/5 ADC1/6 ADC1/7 UART1_TX UART6_RX UART1_TX UART3_TX UART2_TX UART3_RX UART3_CTS
CAN1_RD CAN1_TD CAN2_TD CAN2_RD SPI2_SSEL SPI2_SCLK SPI1_SCLK SPI1_MISO SPI1_MOSI I2C1_SCL I2C3_SCL I2C2_SCL UART3_TX SPI4_SCLK SPI4_MOSI SPI4_MISO UART1_RX ADC1/2 ADC1/3 UART2_RX UART3_RTS SPI6_MOSI UART6_TX
DAC1/2
UART6_TX UART1_RTS UART1_CTS ADC1/15 UART3_TX UART7_TX ADC3/8 UART7_RX ADC1/14 ADC3/15 ADC3/14 UART8_RX UART6_RTS ADC3/9
PWM3/3 PWM3/1 PWM1/4 PWM2/4 PWM1/3N PWM1/2N PWM1/1N PWM1/1 PWM1/2 PWM4/3 PWM4/4
SERIAL_TX CAN1_TD CAN1_RD CAN2_RD UART3_RX ADC1/9 LED3 SPI2_SSEL I2C2_SDA SPI2_MOSI UART3_RTS SPI2_MISO SPI2_SCLK CAN2_TD SPI4_SSEL SPI5_MOSI SPI6_SSEL
PC_8 PC_6 PC_5 U5V PD_8 PA_12 PA_11 PB_12 PB_11 GND PB_2 PB_1 PB_15 PB_14 PB_13 AGND PC_4 PF_5 PF_4 PE_8 PF_10 PE_7 PD_14 PD_15 PF_14 PE_9 GND PE_11 PF_3 PF_15 PF_11 PE_0 PG_8 PG_5 PG_6


CN7
life.augmented
NUCLEO-F429ZI
ZIO HEADER
(top right side)
CN10
PC_6 PB_15 PB_13 PB_12 PA_15 PC_7 PB_5 PB_3 PA_4 PB_4 AVDD AGND GND PB_1 PC_2 PF_4 PB_6 PB_2 GND PD_13 PD_12 PD_11 PE_2 GND PA_0 PB_0 PE_0
UART6_TX SPI2_MOSI PWM1/3N SPI2_SCLK PWM1/1N SPI2_SSEL SPI3_SSEL PWM2/1 UART6_RX PWM3/2 SPI1_MOSI SPI1_SCLK PWM2/2 SPI1_SSEL SPI1_MISO PWM3/1 ADC1/9 ADC1/12 SPI2_MISO ADC3/14 UART1_TX PWM4/1 UART3_RTS UART3_CTS SPI4_SCLK UART4_TX PWM2/1 ADC1/8 PWM1/2N UART8_RX
CAN2_TD CAN2_RD DAC1/1 ADC1/4 I2C1_SCL UART3_CTS UART2_CTS ADC1/0 LED1
PWM4/3 PWM4/4 PWM2/1 PWM3/1 PWM1/1N PWM4/3 PWM4/4 PWM1/1 PWM1/2 PWM1/3 PWM1/2N PWM1/3N PWM1/4 PWM2/3 PWM2/4
I2C1_SCL I2C1_SDA SPI2_SSEL CAN1_RD CAN1_TD SPI1_SCLK ADC1/5 DAC1/2 SPI1_MISO ADC1/6 SPI1_MOSI ADC1/7 SPI4_SSEL SPI4_MISO SPI6_MOSI UART6_TX UART6_RX UART7_TX UART7_RX SPI4_SCLK SPI4_MOSI I2C2_SCL UART3_TX I2C2_SDA UART3_RX
D15 D14 AVDD GND D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
PB_8 PB_9 PA_5 PA_6 PA_7 PD_14 PD_15 PF_12 PF_13 PE_9 PE_11 PF_14 PE_13 PF_15 PG_14 PG_9 PE_8 PE_7 GND PE_10 PE_12 PE_14 PE_15 PB_10 PB_11