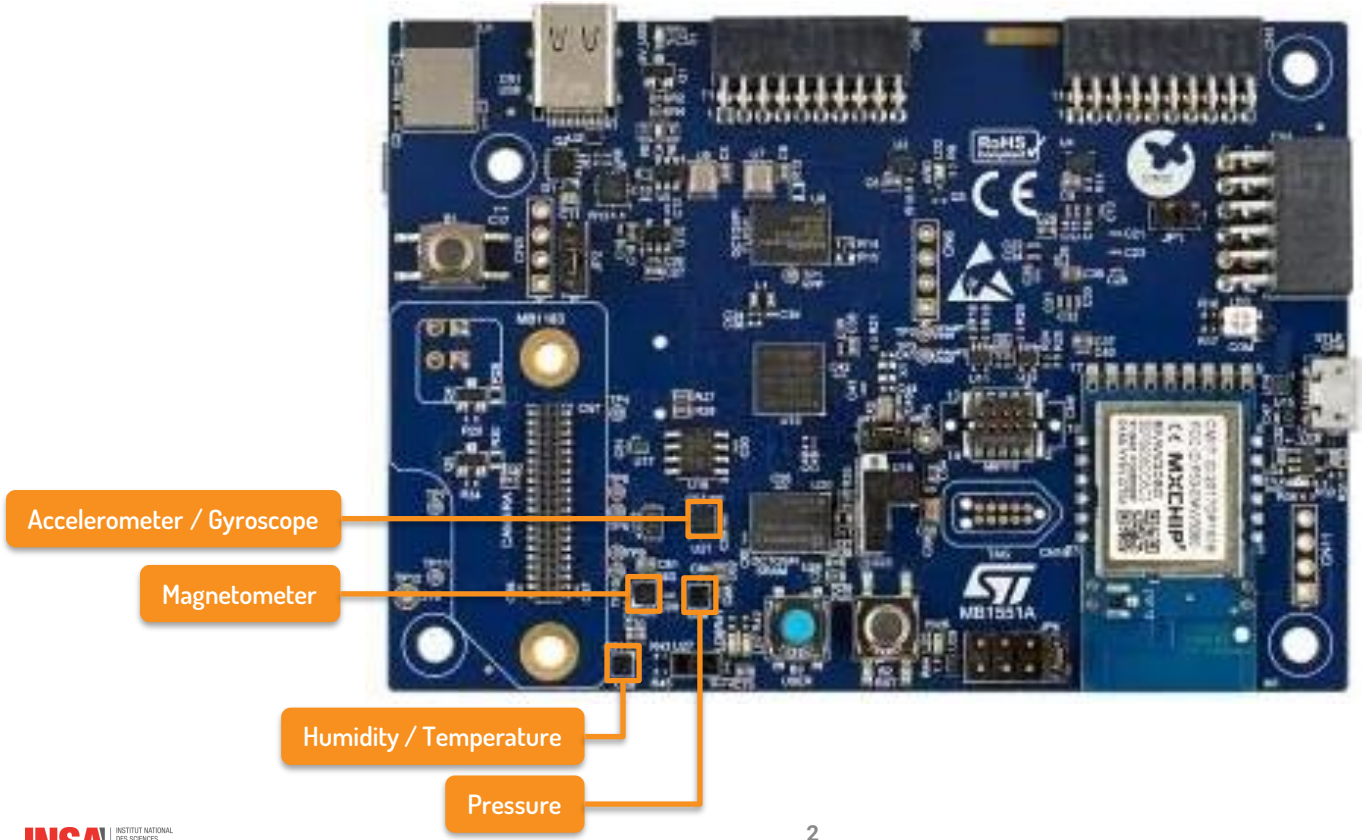
INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

GEA-4-IF3: Project

bertrand.massot@insa-lyon.fr

membre de UNIVERSITÉ DE LYON

Leverage the sensors embedded of the board



Project Dashboard

Connexion

```
Hello World !
Hello World !
Hello World !
Hello World !
Hello World !
Hello World !
Hello World !
```

CONNECTER

Baudrate: 115200

Boussole

Orientation

HTS221

Humidité

0.0%

Température

0.0°C

LPS22HH

Pression

0 mbar

Température

0.0°C

IIS2MDCTR

Magnétomètre

x (mGauss)	y (mGauss)	z (mGauss)
0	0	0

ISM330DHX

Accéléromètre

x (mG)	y (mG)	z (mG)
0	0	0

Gyromètre

x (mdps)	y (mdps)	z (mdps)
0	0	0

Altitude

0.00 m

Project Setup

- **Download, unzip and import the base project on Moodle**
- **Download, unzip and open the dashboard with Chrome or Edge**

Project Setup

main.c file is configured to call project_main() function in project.c

```
/* USER CODE BEGIN Includes */
#include "project.h"
/* USER CODE END Includes */

[…]

int main(void)
{
  /* USER CODE BEGIN 2 */
  project_main();
  /* USER CODE END 2 */

[…]
```

Build and test that everything is working using the web application.

Project Instructions

- **All of your code must be located in project.c**
  - Except for interrupt calls from stm32u5xx_it.c
- **The .ioc file can be modified in CubeMX freely**
- **In project.c, an example of acquiring and sending sensor values is shown**
- **Don’t use HAL_UART_Transmit directly, but rather the functions from protocol.h**

Project objectives

Project objectives

Primary objectives from lower to higher priority:

- Temperature and humidity from HTS221 must be sent at 1 Hertz to UART
- Pressure and temperature from LPS22HH must be sent at 10 Hertz to UART
- Magnetometer axes from IIS2MDCTR must be sent at 20 Hz to UART
- Accelerometer and gyroscope axes from ISM330DHCX must be sent at 100 Hertz to UART

Project objectives

Secondary objectives:

- A press on the USER button start a calibration procedure:
  - A message is sent to UART
  - After one second, gyroscope axes are sampled during 5 seconds at 100 Hertz and averaged to get offset
  - A new message is sent to UART with averaged values
  - Red LED is on during all this procedure
- Implement a model to calculate altitude from LPS22HH sensor using default pressure at sea level

Project objectives

Tertiary objectives:

- Calibrate magnetometer to send normalized values for **x** and **y** axes
  - Normalized values are comprised between -1 and +1
  - Calibration must be made far from electromagnetic sources and metal objects
- Filter magnetometer values using a higher sampling rate to remove noise

Examples of low pass filters

Low pass filter implementation

Simple average filter (FIR filter)

```
/* Define size of the average window */
#define N 15
/* Function taking new sample and returning updated average */
float moving_average_filter(float sample) {
  static float buffer[N] = {0};
  static int index = 0;
  static float average = 0;
  average -= buffer[index]; // substract oldest sample from average
  buffer[index] = sample / N; // store sample in buffer
  average += buffer[index]; // add sample to average
  index ++;
  if (index >= N) index = 0;
  return average;
}
```

Low pass filter implementation

Exponential filter (IIR filter)

```
/* Define weight of new sample */
#define alpha 0.2f
/* Function taking new sample and returning updated average */
float exponential_filter(float sample) {
static float average = 0;
average = ((1.0 - alpha) * average) + (alpha * sample);
return average;
}
```