ACCELEMOTER PROJECT. AUTO-GUIDE.
February, Summer of 2019.

By me.For me();

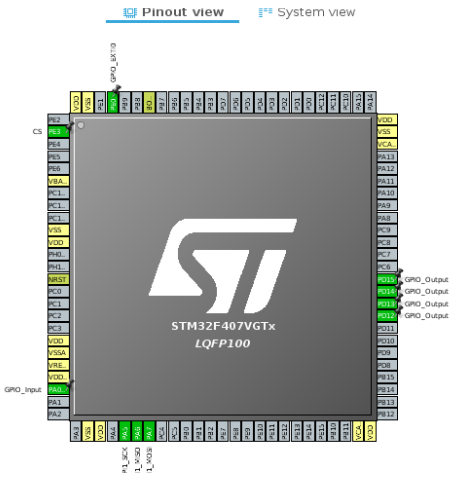
(Worked on Ubuntu 18.04Lts)

1. AttolicWorkspace + StmCubeMX for LIS3DSH sensor.

Let's suppose you remember how to use the software of this section. If you don't know this, visit the course in udemy.com and youtube(and more youtube).

a) List of pins. StmCubeMX (* = obligatory requirement for working)

- Four PinLeds(*)
- One SPI Module(*)
- ResetButton



Next, you can press "Generate Code"

b) Modification of *main.c* file

- First, you need to agree a extra library: `#include "MY_LIS3DSH.h"` and **add a path for this file** (File → Proprieties → search "path" → select a directory for add).
- In the begging of *main section* add `LIS3DSH_InitTypeDef MyAccConfDeff;` `for establish a configurations of the accelerometer.`

- Just up of while(1) loop you need add a configuration parameters for accelerometer

```
/* CONFF PARAMETERS */
MyAccConfDeff.dataRate = LIS3DSH_DATARATE_12_5;
MyAccConfDeff.fullScale = LIS3DSH_FULLSCALE_4;
MyAccConfDeff.antiAliasingBW = LIS3DSH_FILTER_BW_50;
MyAccConfDeff.enableAxes = LIS3DSH_XYZ_ENABLE;
MyAccConfDeff.interruptEnable = false;

LIS3DSH_Init(&hspi1, &MyAccConfDeff);
```

- In the while(1) loop, for initialize a sensor mesure add the next condition:

```
if(LIS3DSH_PollDRDY(1000) == true ){
myData = LIS3DSH_GetDataScaled();   //take a measures of all axis

“rest of the code...”}
```

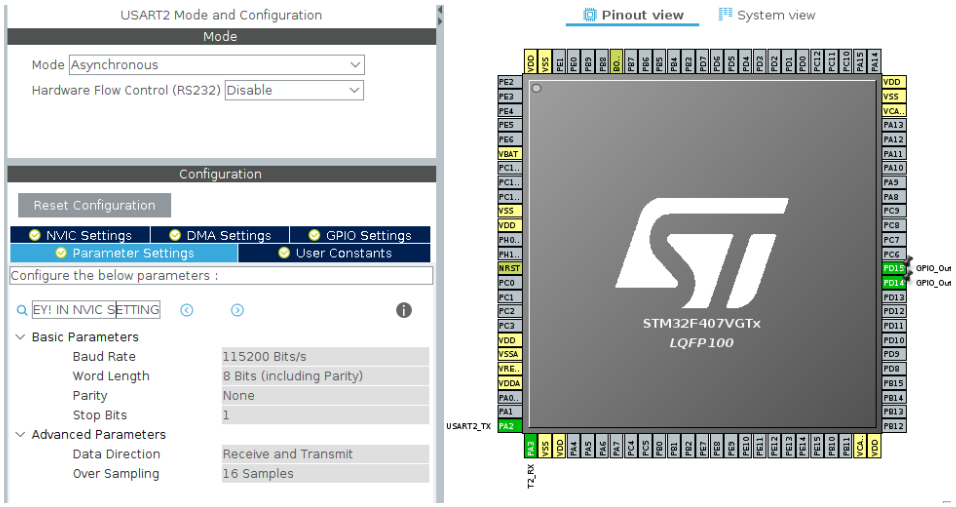
- For access a data measure of diferent axis write a **myData.x** | **myData.y** | **myData.z**
- When you have the rest of your code compile(and pray), next press the debugger icon and press run.

2. AttolicWorkspace + StmCubeMX for UART communication

a) List of pins. StmCubeMX (* = obligatory requirement for working)

- Two UART modules (*)
- Two leds for see the working

pd: in the configuration of UART you need the see nvic and check the boxes (global interrupt)



b) Modification of *main.c* file

- Add a extra libraries (extra PATH is not necessary in this case)

```
#include "stm32f4xx_hal.h"
#include <string.h>
```

- Define your data after in the *main section. For example*

```
char TxData1[30] = "15x\n";
```

- In the while(1) loop you can send the data. For example

```
HAL_UART_Transmit(&huart2, (uint8_t *)TxData1, strlen(TxData1), 5);
```

In this case we send TxData1 from huart2.

3. UART Hadware connections

You need to have a ttl-usb conversor

In the case of the photography of below I use a UART2 module and connect PA2(Tx) and PA3(Rx).

Connections from Stm32f4 to TTL-Usb conversor: PA2(Tx) → Rx pin
PA3(Tx) → Tx pin

You can use 5v or 3v pins of TTL-Usb conversor it as a source power of the StmBoard(in this case same connect GND pin) or you can use usb conector and not use the pins of the TTL-Usb conversor (in this case GND is not necessary)

NOW YOU CAN RUN DE UART CODE AND READ SERIAL PORT WITH SOME SOFTWARE.