



UNIVERSIDAD
DE GRANADA

Máster Universitario en Estructuras
Curso 2020-2021

Práctica de laboratorio: Identificación Modal Operacional de marco experimental

Módulo: MÓDULO FUNDAMENTAL: CALIDAD Y DAÑO

Materia: Análisis Modal y Detección de Defectos

Enrique García Macías

enriquegm@ugr.es

**Departamento de Mecánica de Estructuras e
Ingeniería Hidráulica**

Desarrollo del curso

		FECHA		HORA	PROFESOR	TEMA	
Clase 1	Lunes	1	febrero	9:30-11:30	EGM	1	Introducción: Análisis modal dentro del marco del mantenimiento de la salud estructural.
Clase 2	Lunes	8	febrero	9:30-11:30	EGM	2	Fuentes de deterioro, patologías estructurales, y tecnologías de monitorización.
Clase 3	Lunes	15	febrero	9:30-11:30	EGM	3	Taller: procesamiento de señales.
Clase 4	Lunes	22	febrero	9:30-11:30	EGM	4	Análisis modal experimental.
Clase 5	Lunes	15	marzo	9:30-11:30	EGM	5	Análisis modal operacional.
Clase 6	Lunes	12	abril	9:30-11:30	EGM	6	Análisis modal operacional automatizado.
Clase 7	Lunes	19	abril	9:30-11:30	EGM	7	Práctica de laboratorio I. Taller: Identificación del daño estructural.
Clase 8	Lunes	26	abril	9:30-11:30	RCT	8	Técnicas de identificación dinámica basadas en análisis modal operacional.
Clase 9	Lunes	26	abril	12:00-14:00	RCT	9	Práctica de laboratorio II: Test de vibración ambiental.
Clase 10	Martes	27	abril	9:30-11:30	RCT	10	Casos de estudio.
Clase 11	Martes	27	abril	12:00-14:00	RCT		Presentación de trabajos.

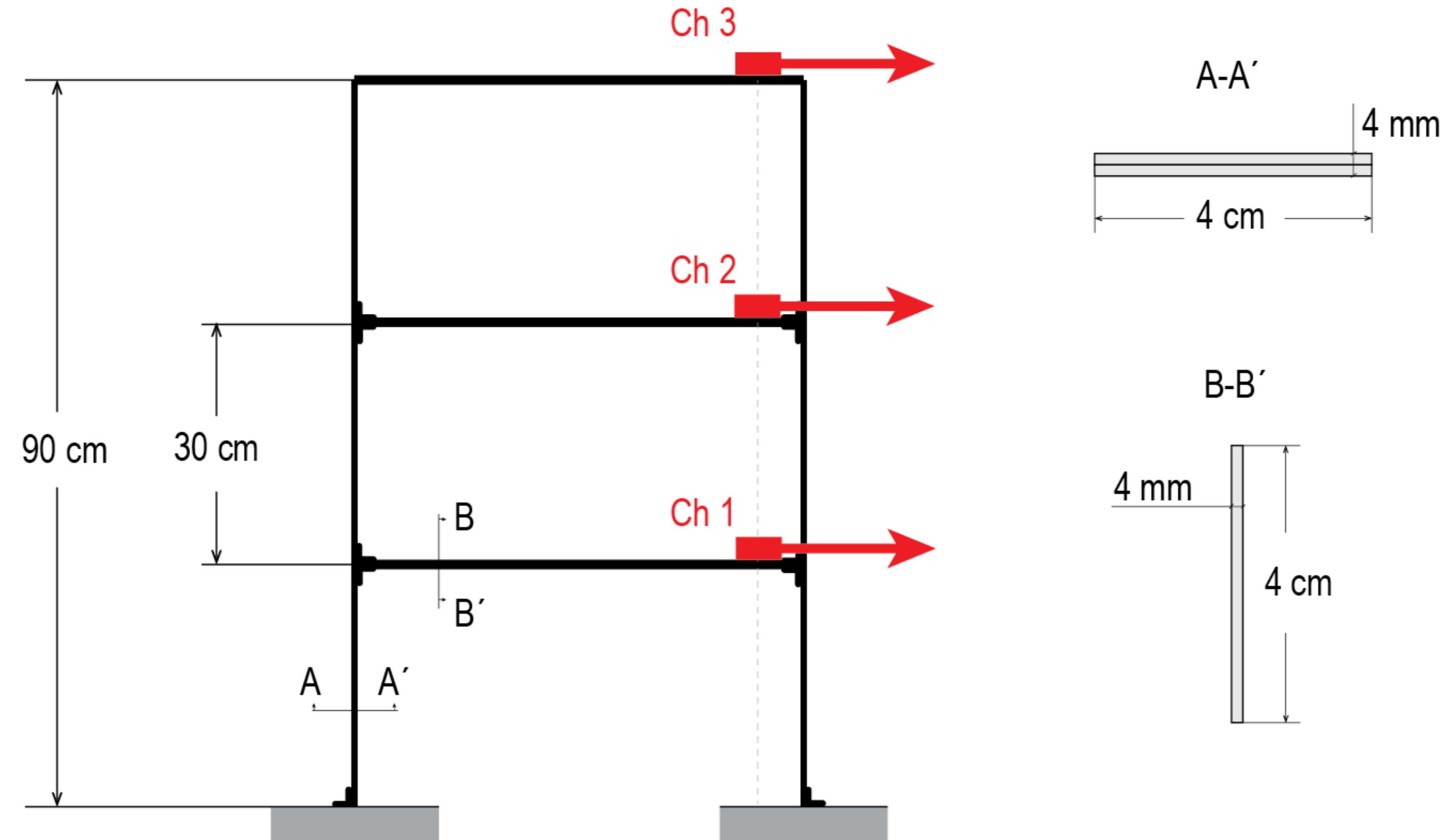
ENTREGA DE TRABAJOS Y EVALUACIÓN

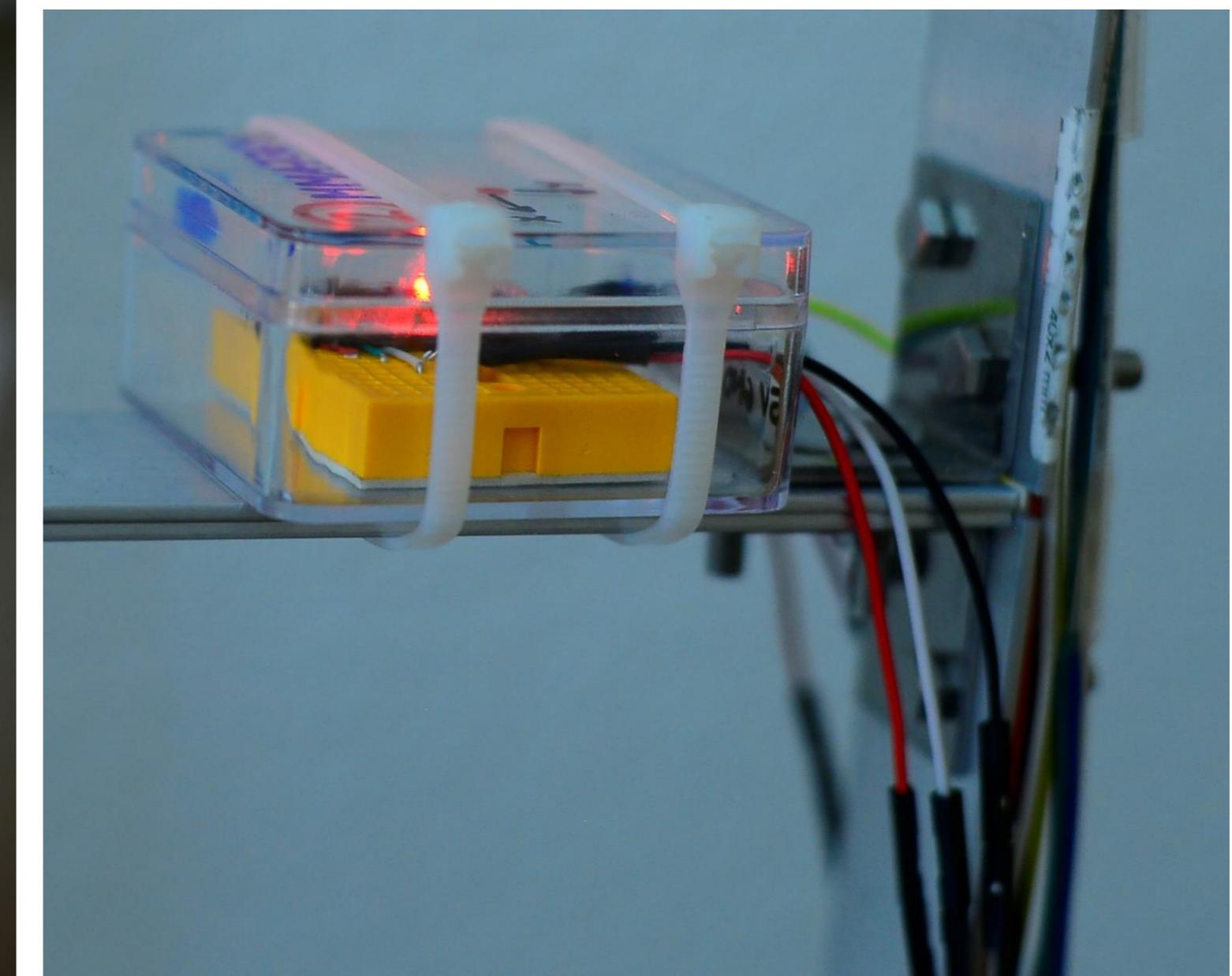
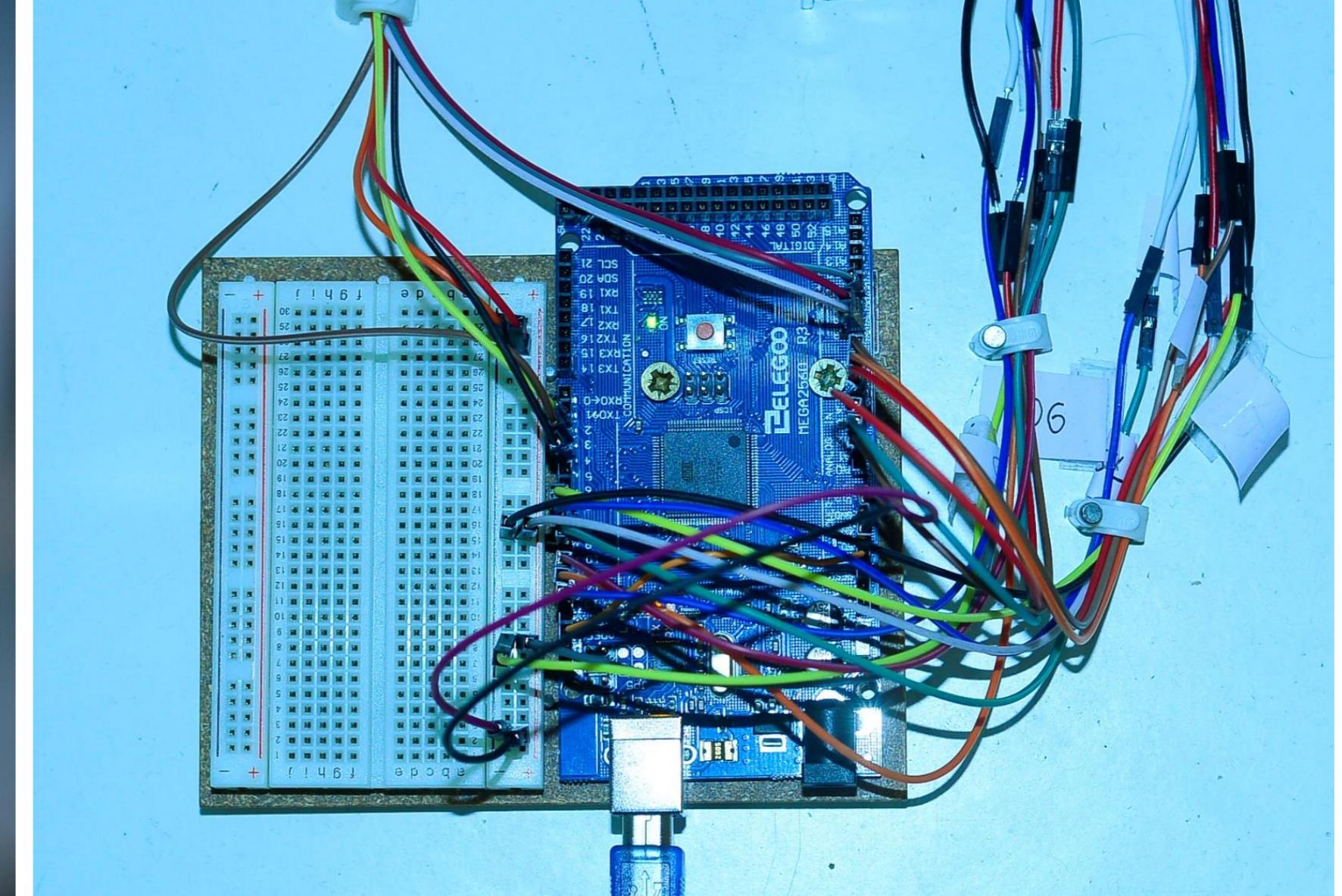
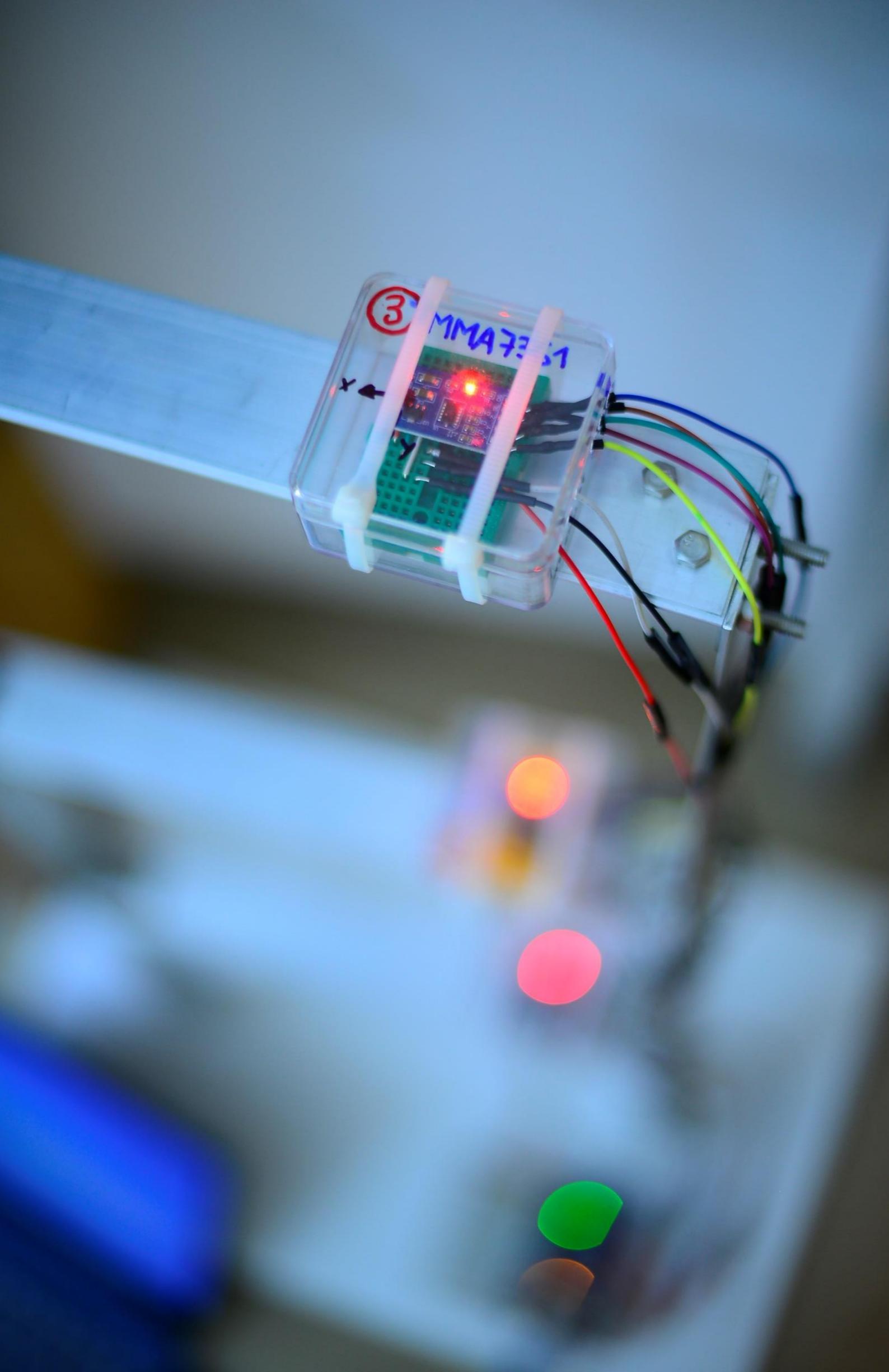
Del 3 al 28 de mayo



UNIVERSIDAD
DE GRANADA

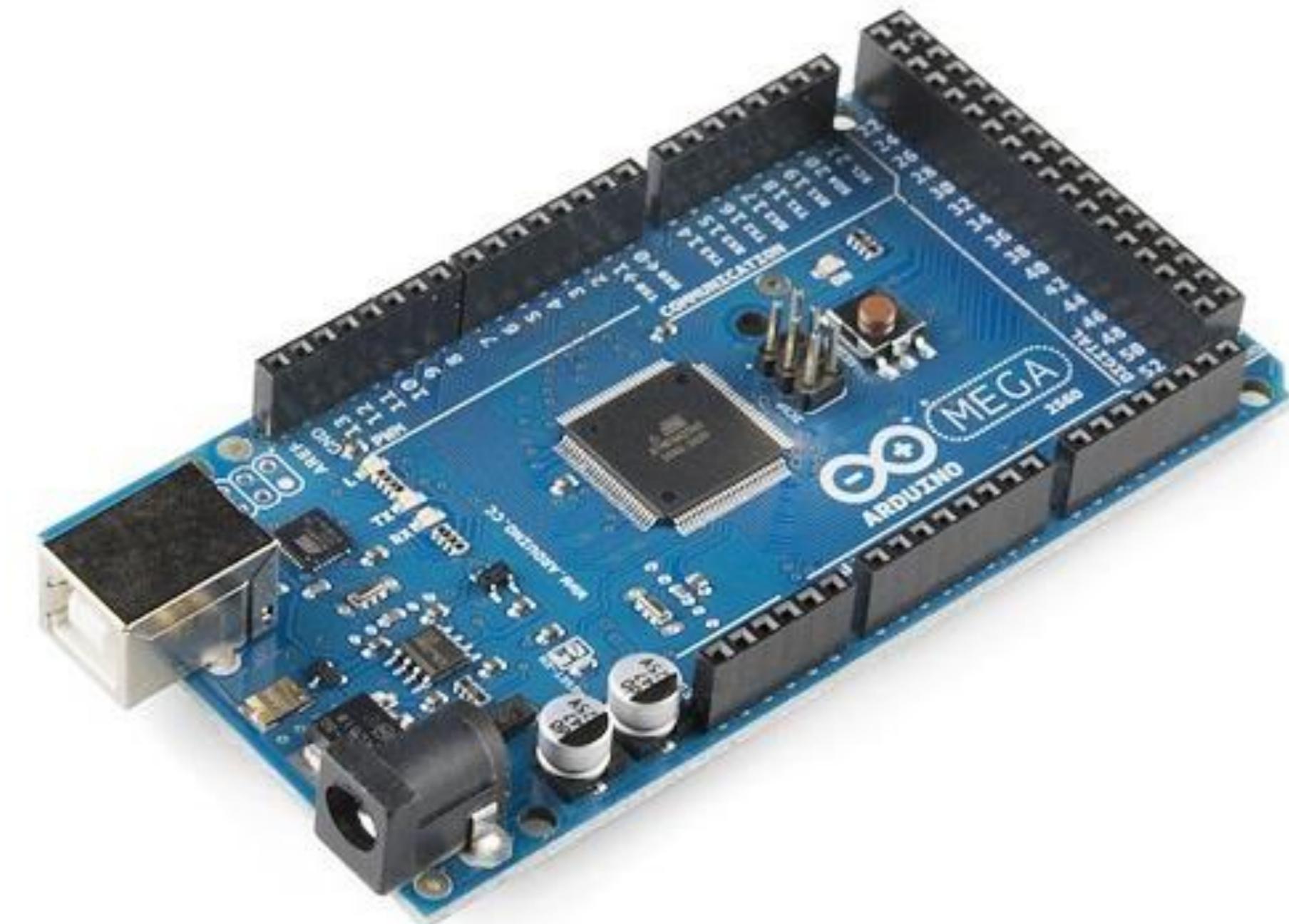
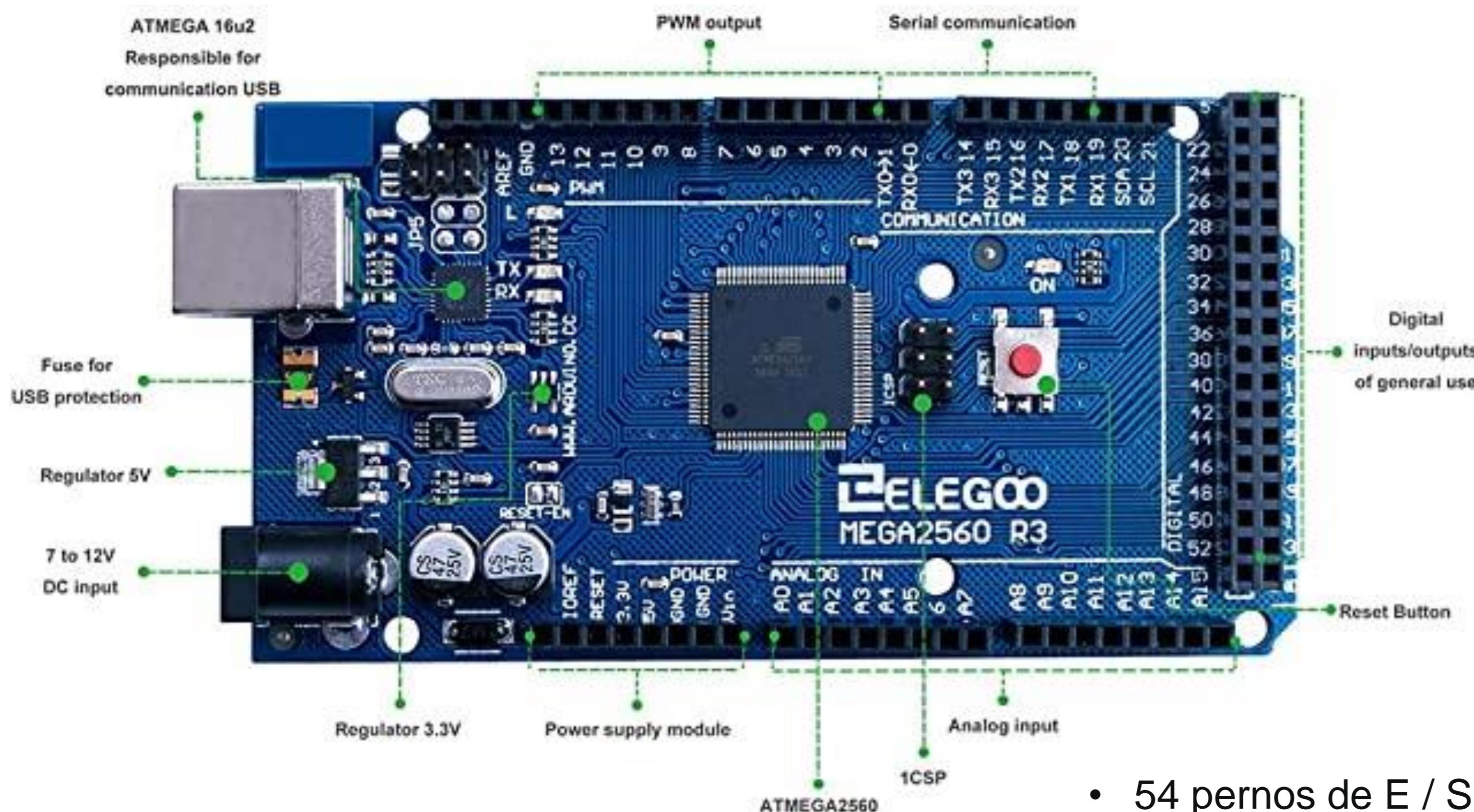
Geometría y set-up experimental





Micro-controlador

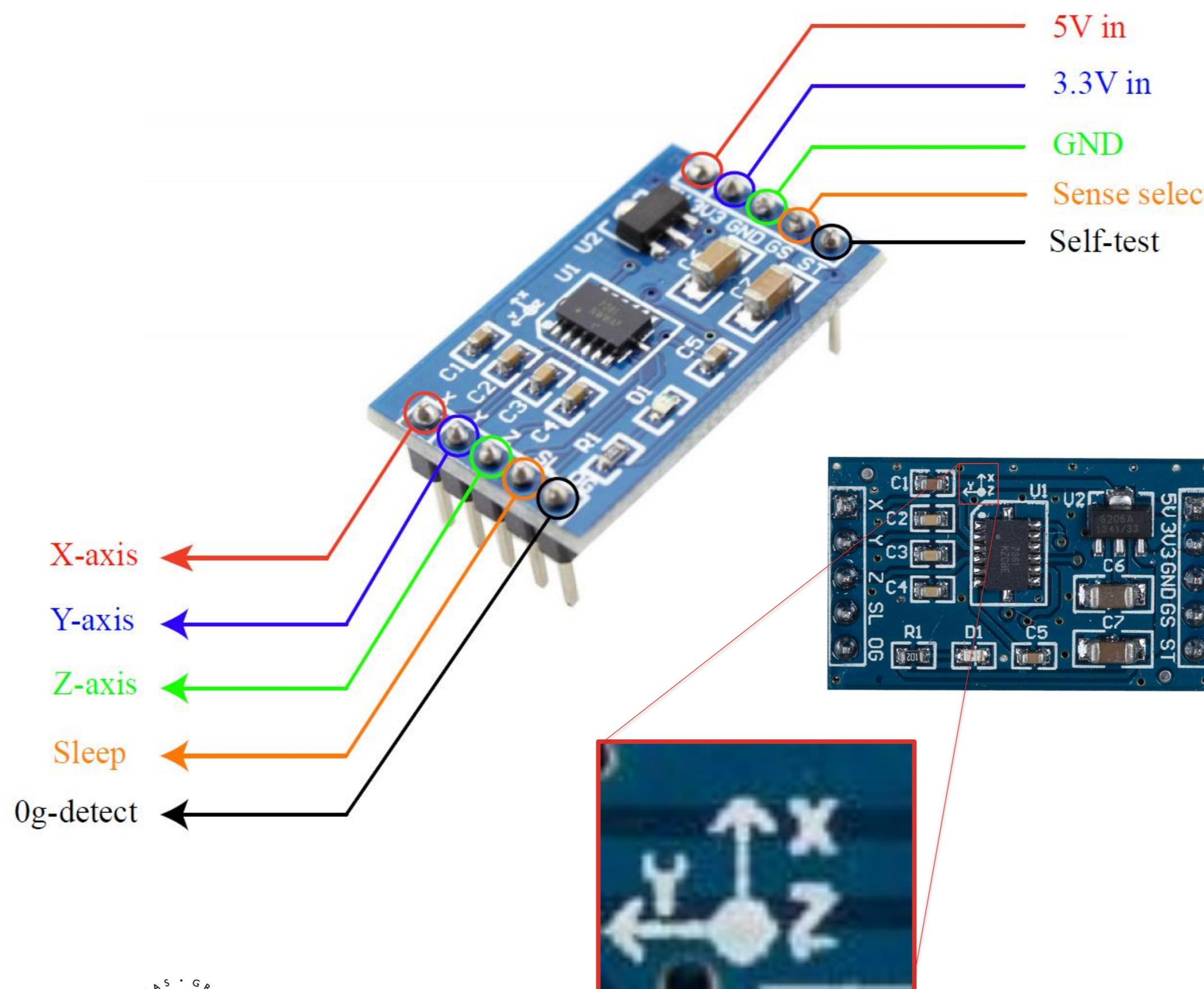
Arduino MEGA



- 54 pernos de E / S digitales y 16 entradas analógicas
- 256 KB de memoria flash para almacenar código
- 10-bits resolution

Acelerómetro

Accelerometer MMA7361



Freescale Semiconductor

Technical Data

Document Number: MMA7361L
Rev 0, 04/2008



±1.5g, ±6g Three Axis Low-g Micromachined Accelerometer

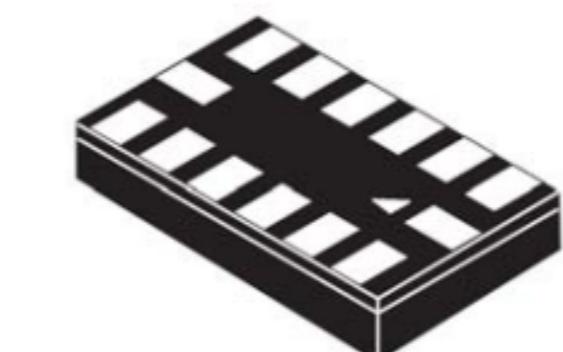
The MMA7361L is a low power, low profile capacitive micromachined accelerometer featuring signal conditioning, a 1-pole low pass filter, temperature compensation, self test, 0g-Detect which detects linear freefall, and g-Select which allows for the selection between 2 sensitivities. Zero-g offset and sensitivity are factory set and require no external devices. The MMA7361L includes a Sleep Mode that makes it ideal for handheld battery powered electronics.

Features

- 3mm x 5mm x 1.0mm LGA-14 Package
- Low Current Consumption: 400 μ A
- Sleep Mode: 3 μ A
- Low Voltage Operation: 2.2 V – 3.6 V
- High Sensitivity (800 mV/g @ 1.5g)
- Selectable Sensitivity (\pm 1.5g, \pm 6g)
- Fast Turn On Time (0.5 ms Enable Response Time)
- Self Test for Freefall Detect Diagnosis
- 0g-Detect for Freefall Protection
- Signal Conditioning with Low Pass Filter
- Robust Design, High Shocks Survivability
- RoHS Compliant
- Environmentally Preferred Product
- Low Cost

MMA7361L:
XYZ AXIS
ACCELEROMETER
 \pm 1.5g, \pm 6g

Bottom View



14 LEAD
LGA
CASE 1977-01

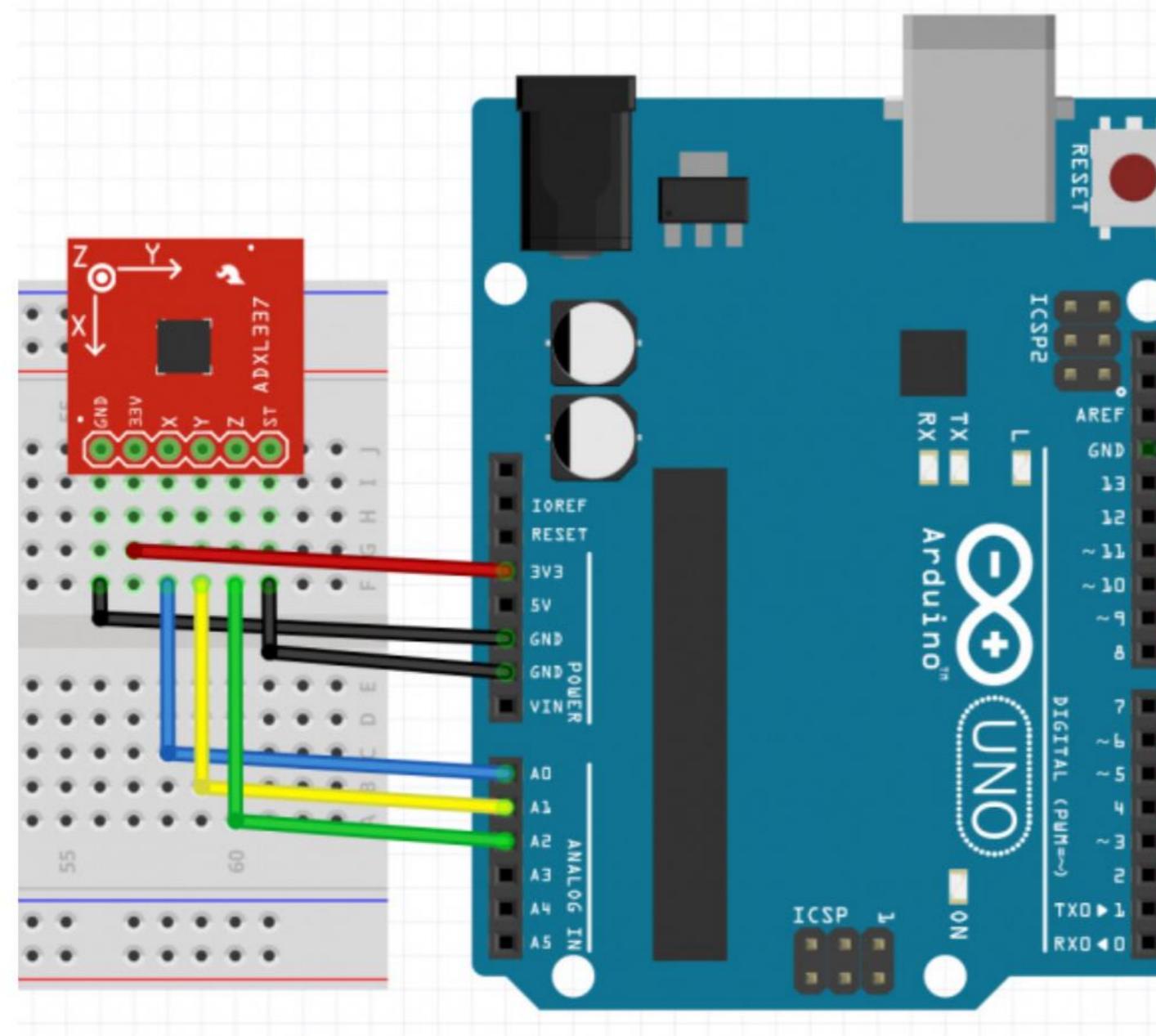
Datasheet



UNIVERSIDAD
DE GRANADA

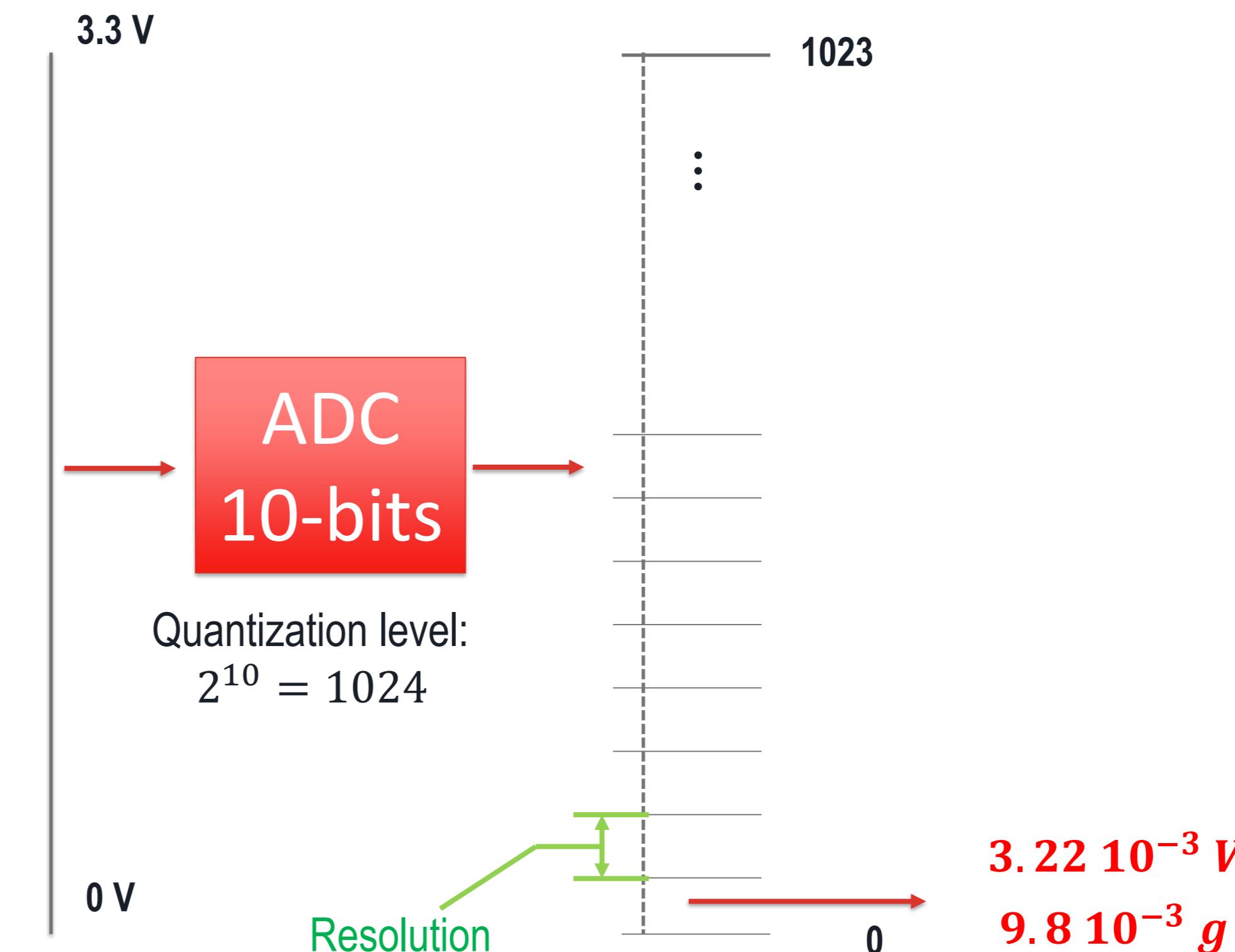
Acelerómetro

Accelerometer MMA7361



Sensitivity
800 mV/g (3,3 V)

$$s_g = \pm 1 \cdot \frac{R}{sens_{sensor}} = \pm 4.025 \cdot 10^{-3} g$$



$$R = \frac{V_{ref}}{N_q - 1} = \frac{3.3}{1024 - 1} V = 3.22 \cdot 10^{-3} V$$

Acelerómetro

Programación acelerómetros

```
#include <AcceleroMMA7361.h>

// EXPERIMENTAL FRAME – 3
ACCELEROMETERS

#include <AcceleroMMA7361.h>

AcceleroMMA7361 acceleroA;
AcceleroMMA7361 acceleroB;
AcceleroMMA7361 acceleroC;

int x; // x axis variable
int y; // y axis variable
int z; // z axis variable
int xb; // x axis variable
int yb; // y axis variable
int zb; // z axis variable
int xc; // x axis variable
int yc; // y axis variable
int zc; // z axis variable
unsigned long time;
```

```
void setup() {
    Serial.begin(115200); // frequency of the measure

    acceleroA.begin(11, 12, 13, 10, A1, A2, A3);
    acceleroA.setARefVoltage(3.3);           //sets the AREF voltage to 3.3V
    acceleroA.setSensitivity(HIGH);          //sets the sensitivity to +/-1.5G
    analogReference(EXTERNAL);

    acceleroB.begin(7, 8, 9, 10, A5, A6, A7);
    acceleroB.setARefVoltage(3.3);           //sets the AREF voltage to 3.3V
    acceleroB.setSensitivity(HIGH);          //sets the sensitivity to +/-1.5G
    analogReference(EXTERNAL);

    acceleroC.begin(2, 3, 4, 10, A8, A9, A10);
    acceleroC.setARefVoltage(3.3);           //sets the AREF voltage to 3.3V
    acceleroC.setSensitivity(HIGH);          //sets the sensitivity to +/-1.5G
    analogReference(EXTERNAL);
}

void loop() {
    time = micros();Serial.print(time);
    Serial.print(" ");

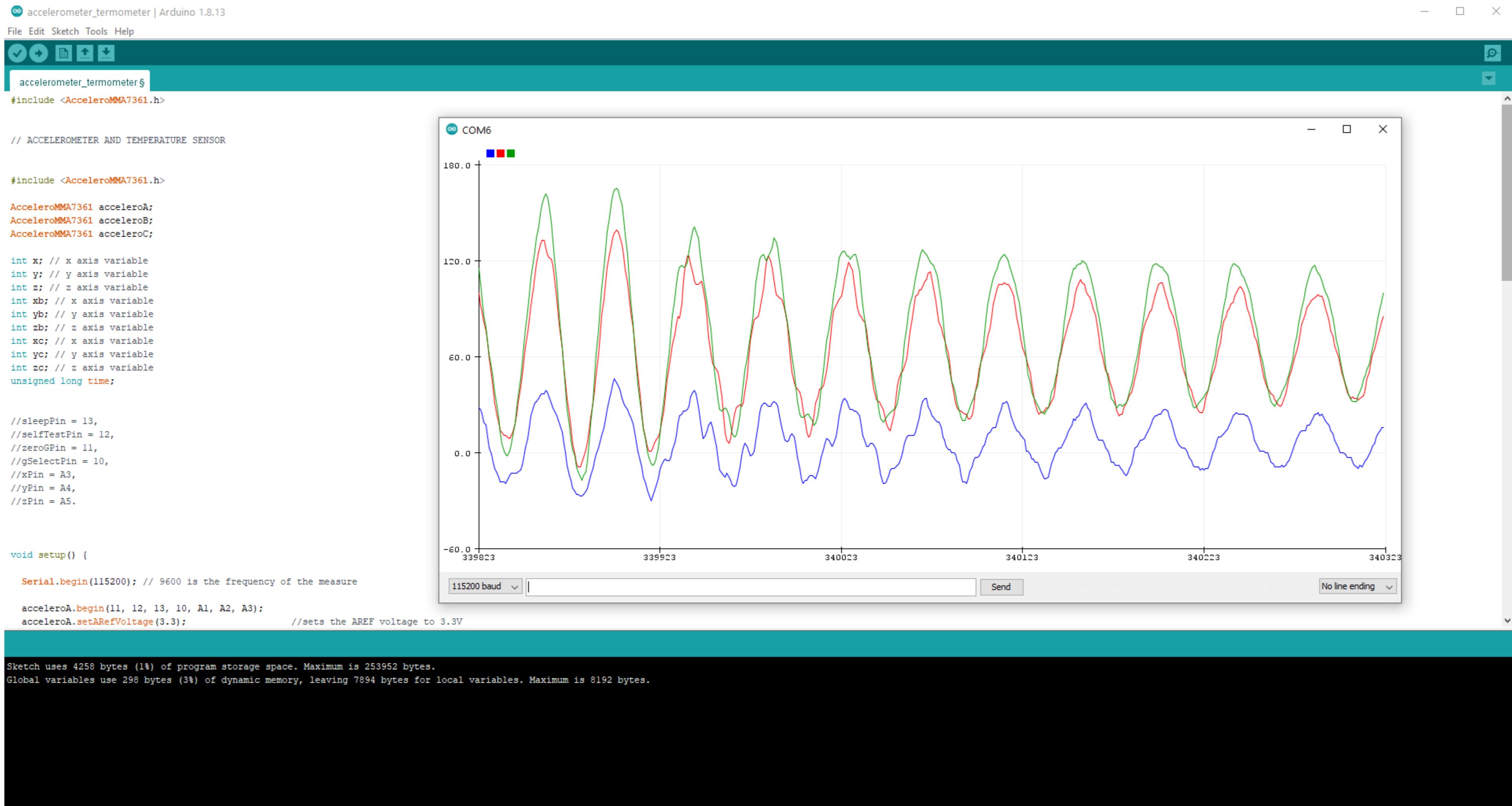
    y = acceleroA.getYAccel();
    yb = acceleroB.getYAccel();
    yc = acceleroC.getYAccel();

    Serial.print(y); // LEVEL 1
    Serial.print(" ");
    Serial.print(yb); // LEVEL 2
    Serial.print(" ");
    Serial.println(yc); // LEVEL 3
}
```

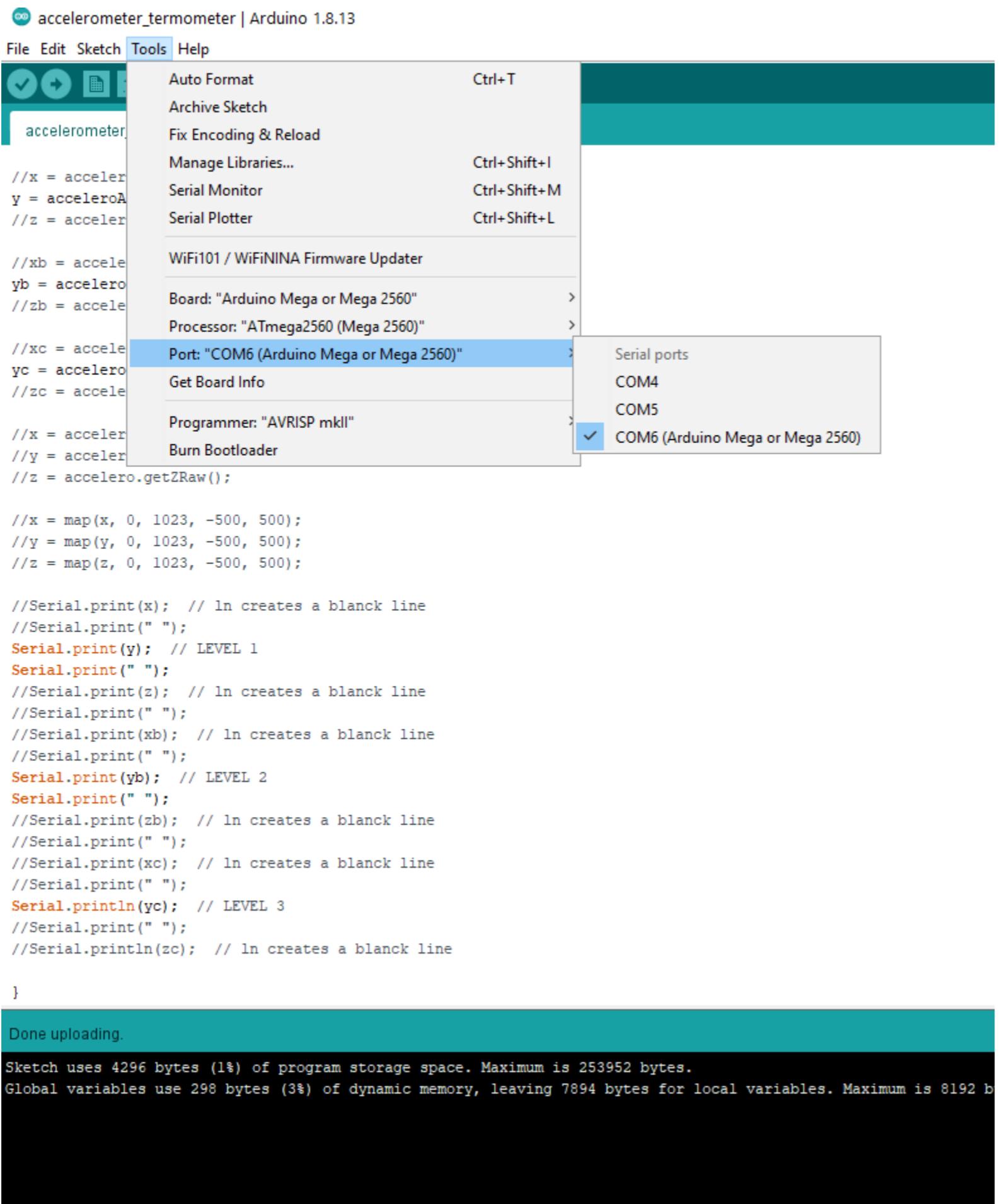


Acelerómetro

Programación acelerómetros



Toma de medidas



The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The Tools menu is open, showing options like Auto Format (Ctrl+T), Archive Sketch, Fix Encoding & Reload, Manage Libraries..., Serial Monitor (Ctrl+Shift+M), Serial Plotter (Ctrl+Shift+L), WiFi101 / WiFiNINA Firmware Updater, Board: "Arduino Mega or Mega 2560", Processor: "ATmega2560 (Mega 2560)", Port: "COM6 (Arduino Mega or Mega 2560)" (which is selected), Get Board Info, Programmer: "AVRISP mkII", Burn Bootloader, and WiFi101 / WiFiNINA Firmware Updater. The main area displays the following C++ code:

```
//accelerometer
//acceleroA
//acceleroB
//acceleroC
//acceleroD
//acceleroE
//acceleroF
//acceleroG
//acceleroH
//acceleroI
//acceleroJ
//acceleroK
//acceleroL
//acceleroM
//acceleroN
//acceleroO
//acceleroP
//acceleroQ
//acceleroR
//acceleroS
//acceleroT
//acceleroU
//acceleroV
//acceleroW
//acceleroX
//acceleroY
//acceleroZ
//acceleroA = analogRead(A0);
//acceleroB = analogRead(A1);
//acceleroC = analogRead(A2);
//acceleroD = analogRead(A3);
//acceleroE = analogRead(A4);
//acceleroF = analogRead(A5);
//acceleroG = analogRead(A6);
//acceleroH = analogRead(A7);
//acceleroI = analogRead(A8);
//acceleroJ = analogRead(A9);
//acceleroK = analogRead(A10);
//acceleroL = analogRead(A11);
//acceleroM = analogRead(A12);
//acceleroN = analogRead(A13);
//acceleroO = analogRead(A14);
//acceleroP = analogRead(A15);
//acceleroQ = analogRead(A16);
//acceleroR = analogRead(A17);
//acceleroS = analogRead(A18);
//acceleroT = analogRead(A19);
//acceleroU = analogRead(A20);
//acceleroV = analogRead(A21);
//acceleroW = analogRead(A22);
//acceleroX = analogRead(A23);
//acceleroY = analogRead(A24);
//acceleroZ = analogRead(A25);
//acceleroA = map(analogRead(A0), 0, 1023, -500, 500);
//acceleroB = map(analogRead(A1), 0, 1023, -500, 500);
//acceleroC = map(analogRead(A2), 0, 1023, -500, 500);
//acceleroD = map(analogRead(A3), 0, 1023, -500, 500);
//acceleroE = map(analogRead(A4), 0, 1023, -500, 500);
//acceleroF = map(analogRead(A5), 0, 1023, -500, 500);
//acceleroG = map(analogRead(A6), 0, 1023, -500, 500);
//acceleroH = map(analogRead(A7), 0, 1023, -500, 500);
//acceleroI = map(analogRead(A8), 0, 1023, -500, 500);
//acceleroJ = map(analogRead(A9), 0, 1023, -500, 500);
//acceleroK = map(analogRead(A10), 0, 1023, -500, 500);
//acceleroL = map(analogRead(A11), 0, 1023, -500, 500);
//acceleroM = map(analogRead(A12), 0, 1023, -500, 500);
//acceleroN = map(analogRead(A13), 0, 1023, -500, 500);
//acceleroO = map(analogRead(A14), 0, 1023, -500, 500);
//acceleroP = map(analogRead(A15), 0, 1023, -500, 500);
//acceleroQ = map(analogRead(A16), 0, 1023, -500, 500);
//acceleroR = map(analogRead(A17), 0, 1023, -500, 500);
//acceleroS = map(analogRead(A18), 0, 1023, -500, 500);
//acceleroT = map(analogRead(A19), 0, 1023, -500, 500);
//acceleroU = map(analogRead(A20), 0, 1023, -500, 500);
//acceleroV = map(analogRead(A21), 0, 1023, -500, 500);
//acceleroW = map(analogRead(A22), 0, 1023, -500, 500);
//acceleroX = map(analogRead(A23), 0, 1023, -500, 500);
//acceleroY = map(analogRead(A24), 0, 1023, -500, 500);
//acceleroZ = map(analogRead(A25), 0, 1023, -500, 500);
//Serial.print(x); // ln creates a blank line
//Serial.print(" ");
Serial.print(y); // LEVEL 1
Serial.print(" ");
//Serial.print(z); // ln creates a blank line
//Serial.print(" ");
//Serial.print(xb); // ln creates a blank line
//Serial.print(" ");
//Serial.print(xc); // ln creates a blank line
//Serial.print(" ");
Serial.print(yb); // LEVEL 2
Serial.print(" ");
//Serial.print(zb); // ln creates a blank line
//Serial.print(" ");
//Serial.print(xc); // ln creates a blank line
//Serial.print(" ");
//Serial.print(yc); // LEVEL 3
Serial.println(yc); // LEVEL 3
//Serial.print(" ");
//Serial.println(zc); // ln creates a blank line
}

```

Done uploading.

Sketch uses 4296 bytes (1%) of program storage space. Maximum is 253952 bytes.
Global variables use 298 bytes (3%) of dynamic memory, leaving 7894 bytes for local variables. Maximum is 8192 b

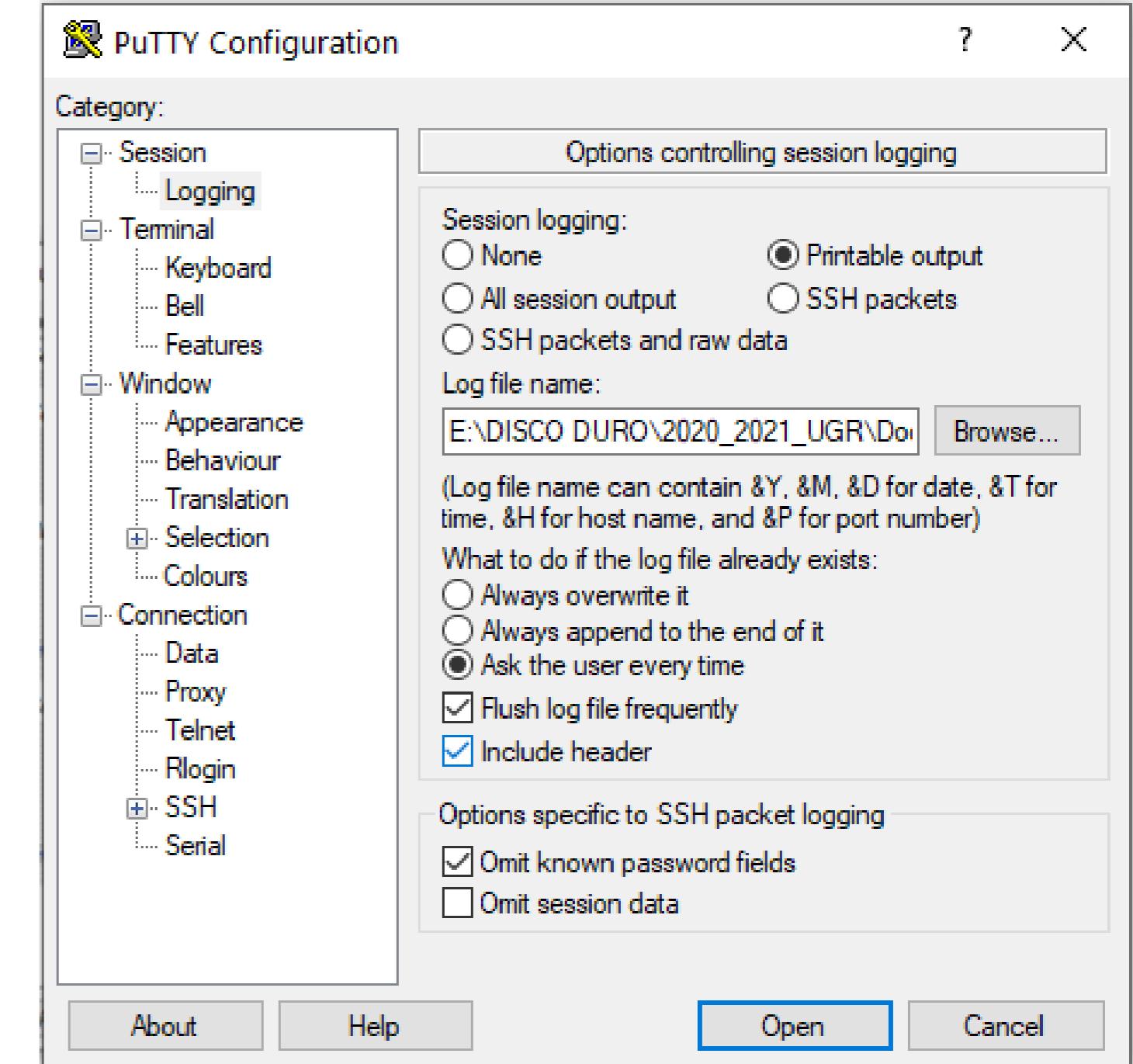
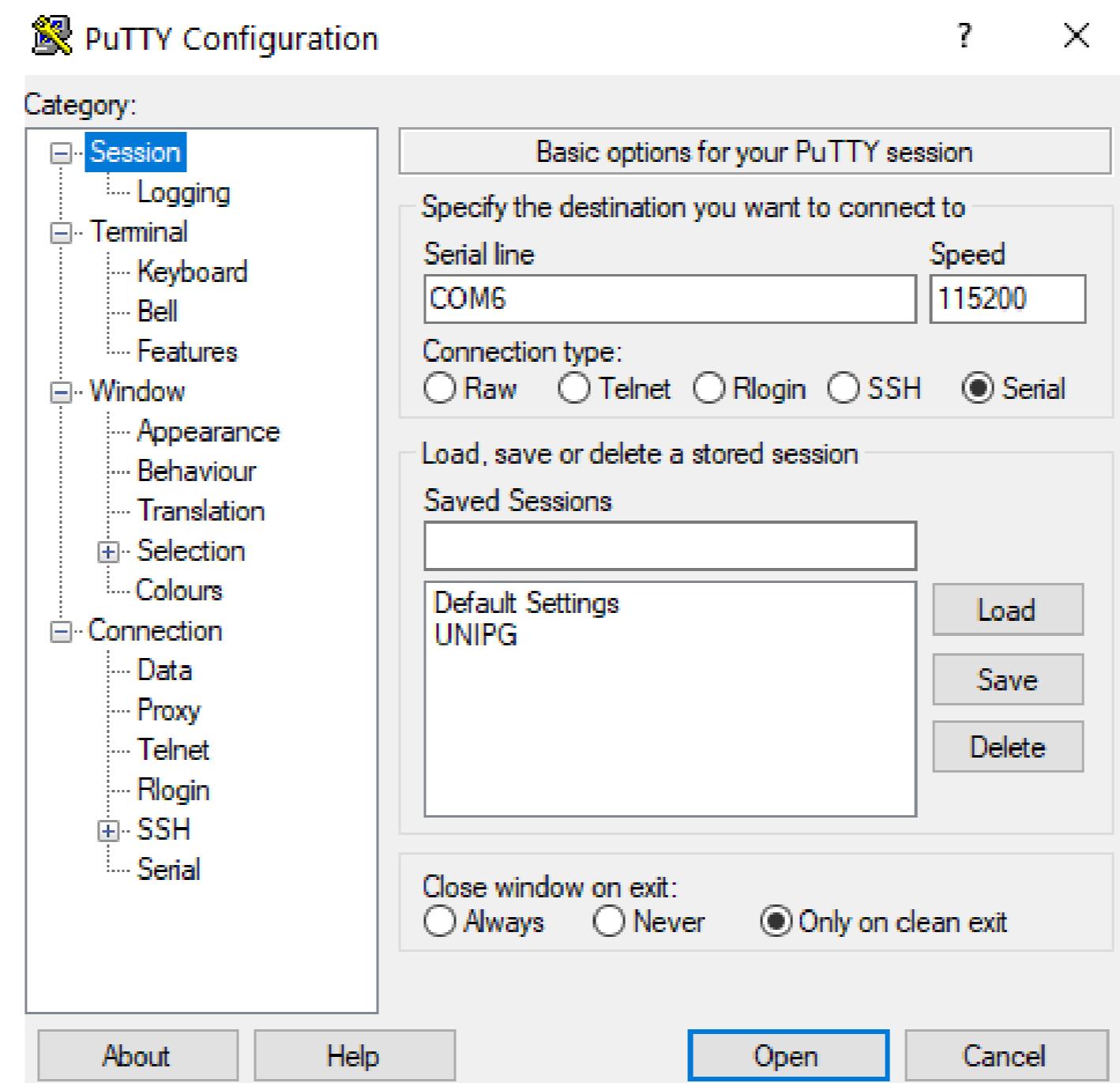
Puerto de conexión



UNIVERSIDAD
DE GRANADA



PuTTY



Toma de medidas

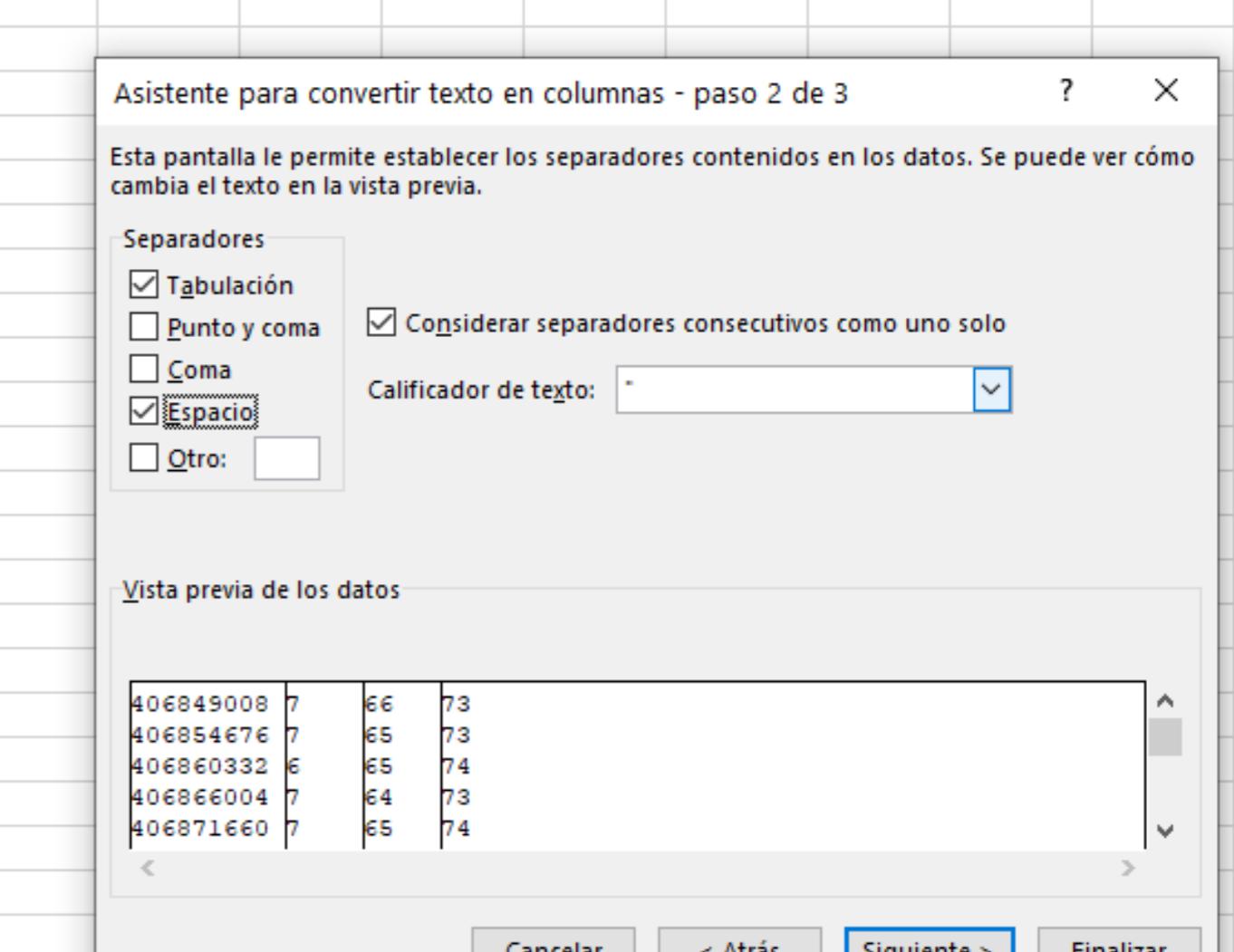
// getXAccel(): Returns the acceleration of the X-axis as a int (1 G = 100.00)

COM6 - PuTTY

```
6732448 -4 41 46
6738064 -1 45 52
6743664 -1 50 58
6749268 3 55 66
6754852 4 59 73
6760420 6 66 80
6766004 8 71 87
6771572 10 77 93
6777212 12 82 99
6782828 16 85 104
6788512 19 90 110
6794184 21 93 113
6799872 23 95 115
6805568 24 98 118
6811248 26 100 118
6816988 25 102 117
6822700 25 101 117
6828420 24 99 116
6834080 22 95 115
6839768 21 91 112
6845464 18 88 106
6851144 16 84 100
6856832 14 81 94
6862460
```

ID	X	Y	Z	Delta t [s]	Time	Acceleration level 1			Acceleration level 2			Acceleration level 3		
						#VALOR!	[s]	[g]	#VALOR!	[s]	[g]	#VALOR!	[s]	[g]
406849008	7	66	73	4	64	74	0.000	0.040	0.640	0.740	0.000	0.000	0.000	
406854676	7	65	73			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406860332	6	65	74			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406866004	7	64	73			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406871660	7	65	74			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406877332	7	65	73			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406882988	7	68	49008	7	66	73		#VALOR!	0.000	0.000	0.000	0.000	0.000	0.000
406854676	7	65	73			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406860332	6	65	74			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406866004	7	64	73			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406871660	7	65	74			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406877332	7	65	73			#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
406882988	7	62	04	42	65	77		#VALOR!	0.000	0.000	0.000	0.000	0.000	0.000
56841065	77					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
11132866	77					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
16612965	76					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
22080965	77					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
27548865	77					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
33028865	77					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
38504965	76					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
43980865	77					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
49460865	76					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
54928765	76					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
60388865	77					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000
65868866	76					#VALOR!		0.000	0.000	0.000	0.000	0.000	0.000	0.000

Ajustar_medidas.xlsx



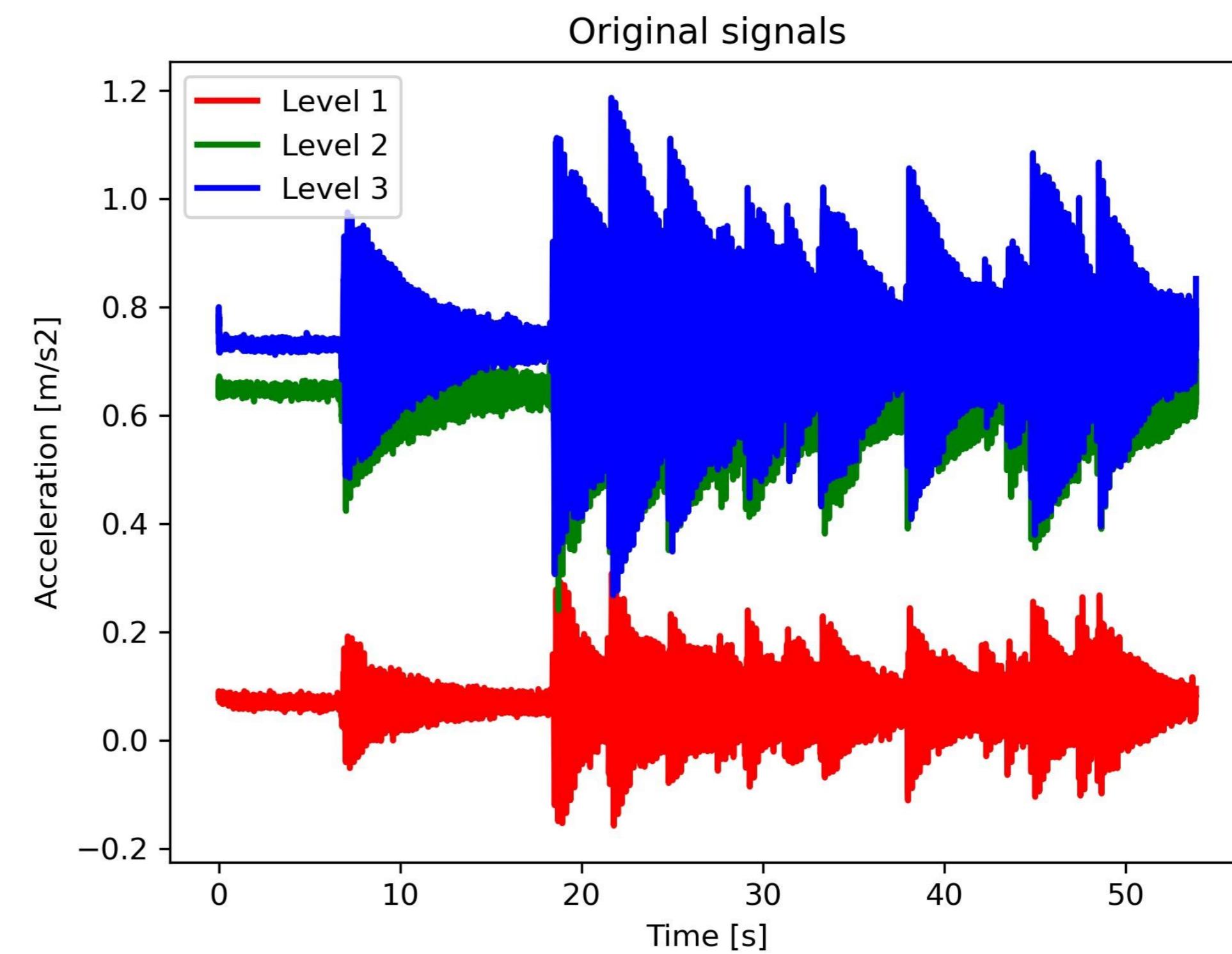
Toma de medidas

Name	Date modified	Type
Ajustar_medidas.xlsx		
Measurement_1.log		
Measurement_1.txt		
~\$Ajustar_medidas.xlsx		

*Measurement_1.txt - Notepad

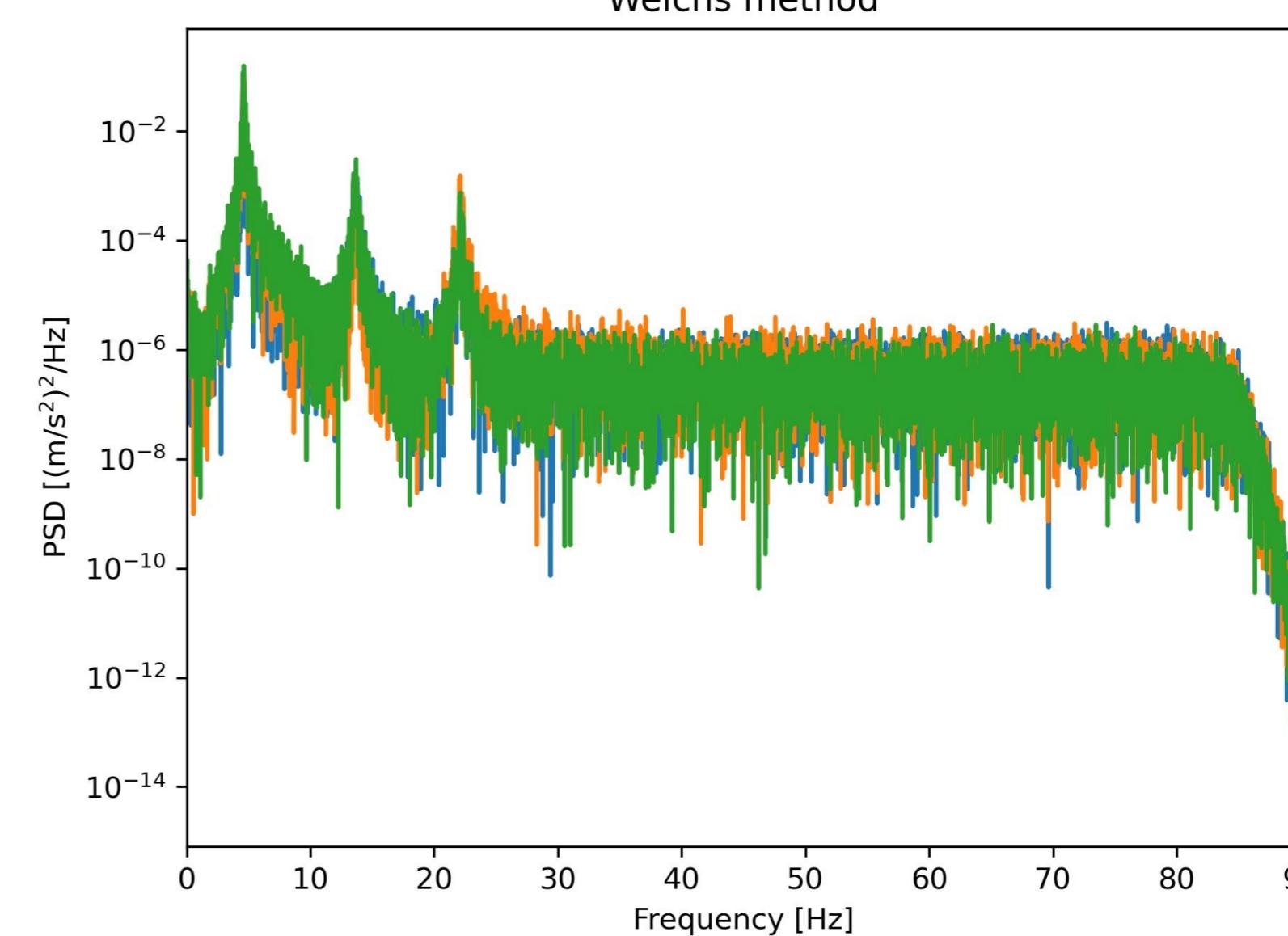
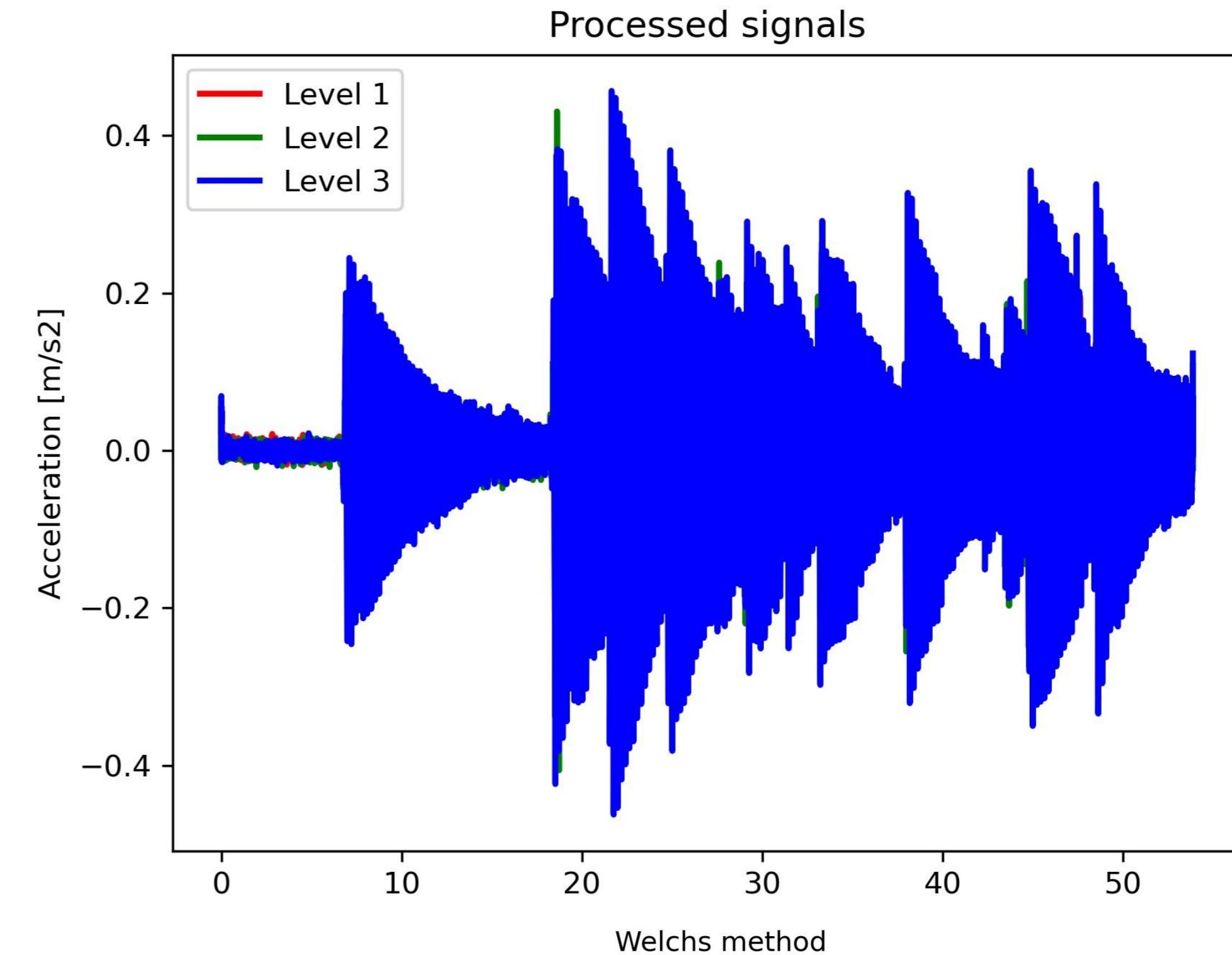
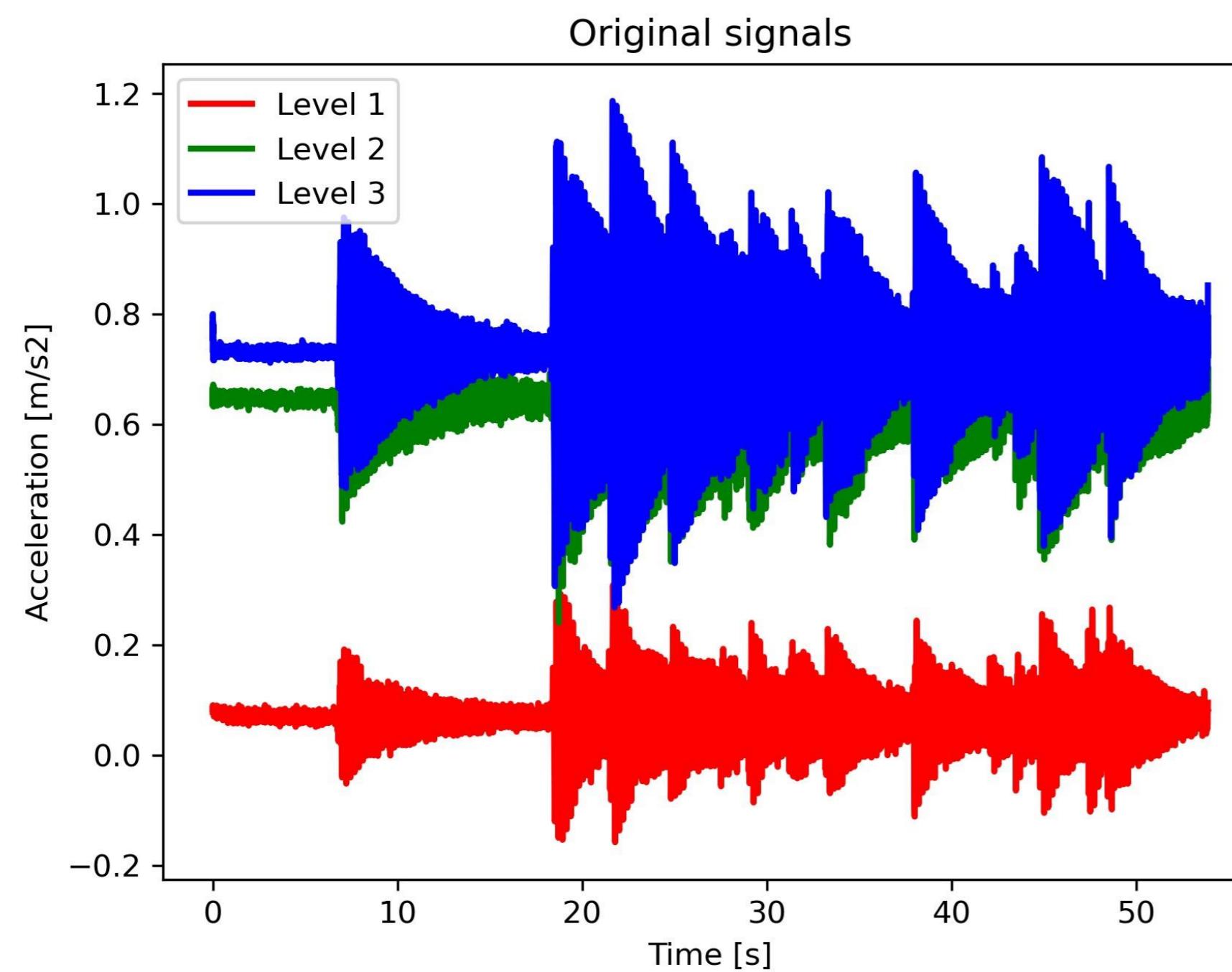
	File	Edit	Format	View	Help
0.000	0.100	0.650	0.770		
0.005	0.080	0.660	0.770		
0.011	0.090	0.650	0.760		
0.016	0.090	0.650	0.770		
0.022	0.080	0.650	0.770		
0.027	0.080	0.650	0.770		
0.033	0.090	0.650	0.760		
0.038	0.080	0.650	0.770		
0.044	0.080	0.650	0.760		
0.049	0.070	0.650	0.760		
0.055	0.080	0.650	0.770		
0.060	0.080	0.660	0.760		
0.066	0.090	0.650	0.780		
0.071	0.080	0.650	0.770		
0.077	0.080	0.650	0.770		
0.082	0.080	0.650	0.780		
0.087	0.090	0.660	0.770		

Generate_measurements.py



Toma de medidas

Generate_measurements.py



UNIVERSIDAD
DE GRANADA

Identificación Modal

```
Eyo = 69*1E+9; # N/m2  
dens = 2700; # kg/m3
```

```
Ar = 0.04*0.002;
```

```
lne = (1/12)*0.04*0.002**3; # m^4
```

```
h = 0.3; # m
```

```
Le = 0.3; # m
```

```
k1 = 2*12*Eyo*lne/h**3;
```

```
k2 = k1;
```

```
k3 = k1;
```

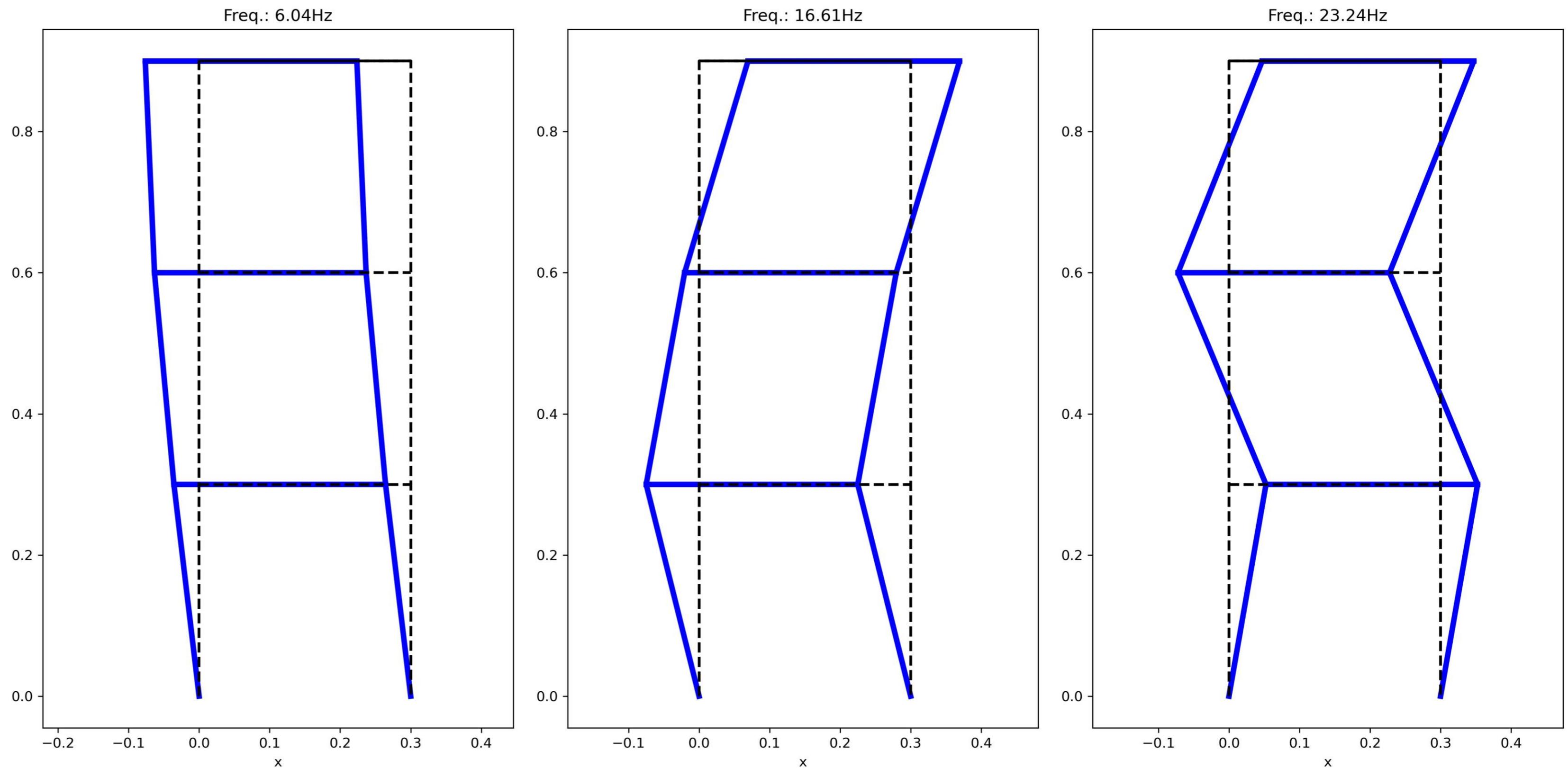
```
m1 = 2*h*Ar*dens+2*Ar*Le*dens;
```

```
m2 = m1;
```

```
m3 = h*Ar*dens+2*Ar*Le*dens;
```

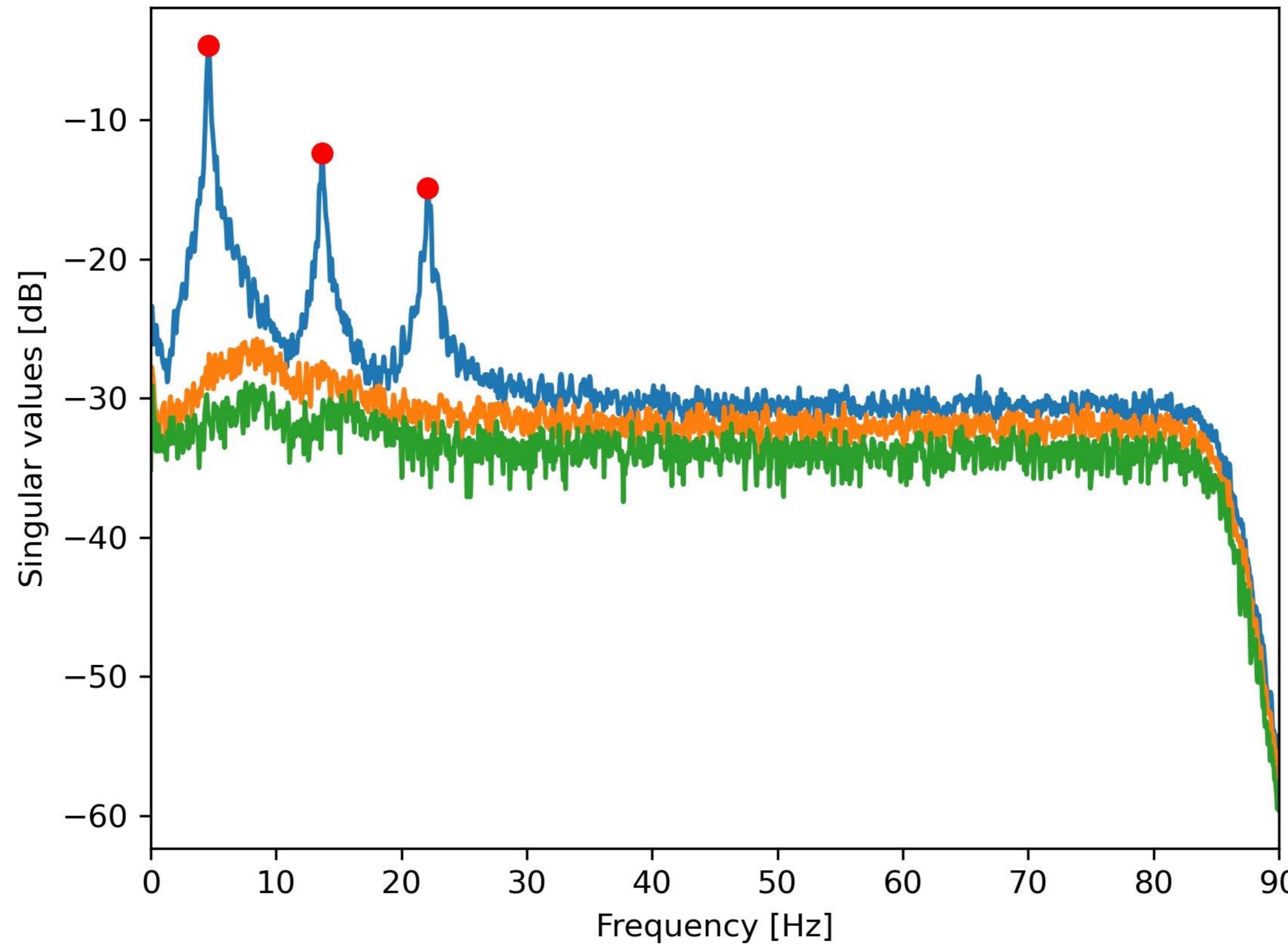
```
M = np.array([[m1,0,0],[0,m2,0],[0,0,m3]])
```

```
K = np.array([[k1+k2,-k2,0],[-k2,k2+k3,-k3],[0,-k3,k3]])
```



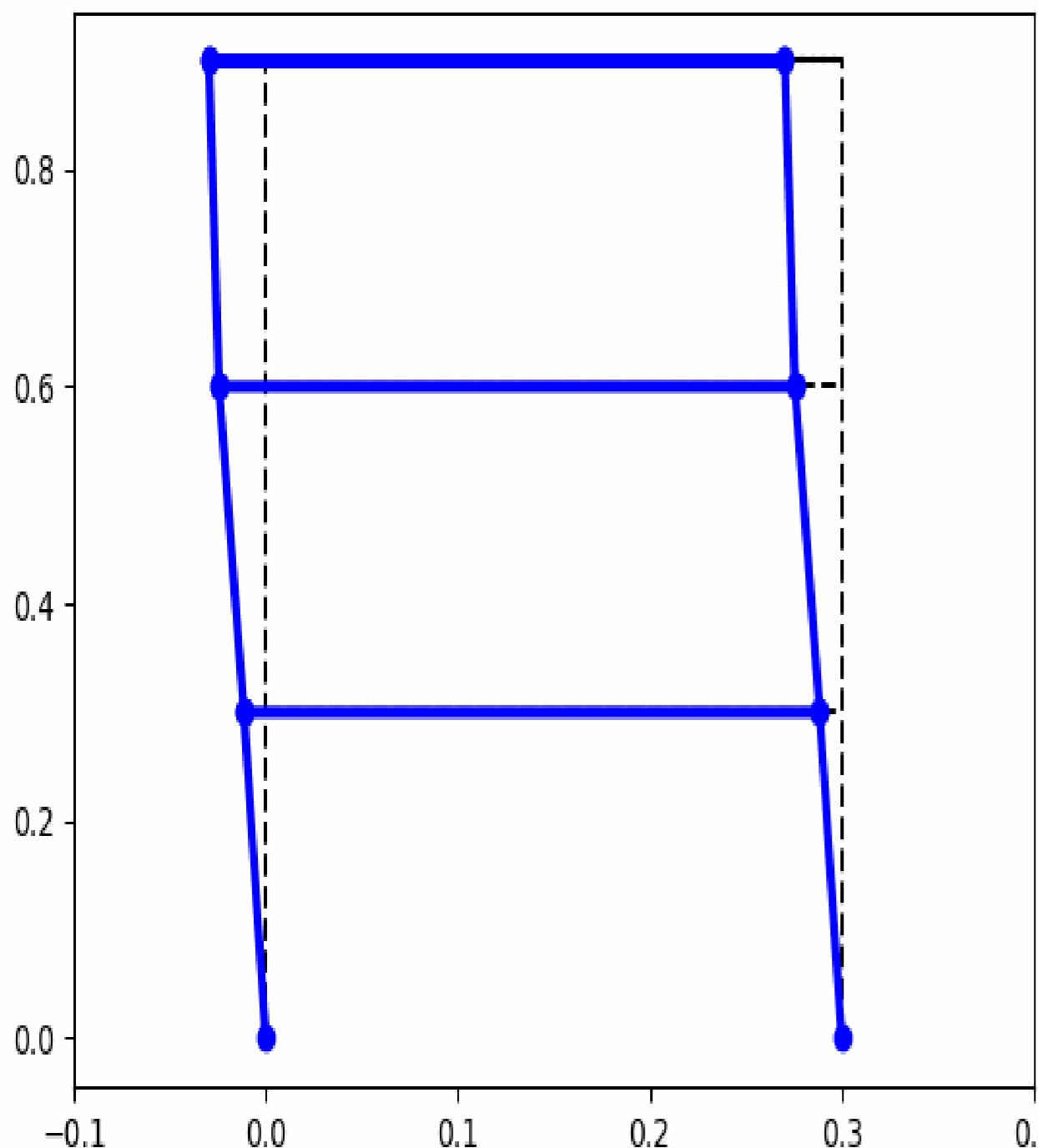
FDD

OMA_FDD.py

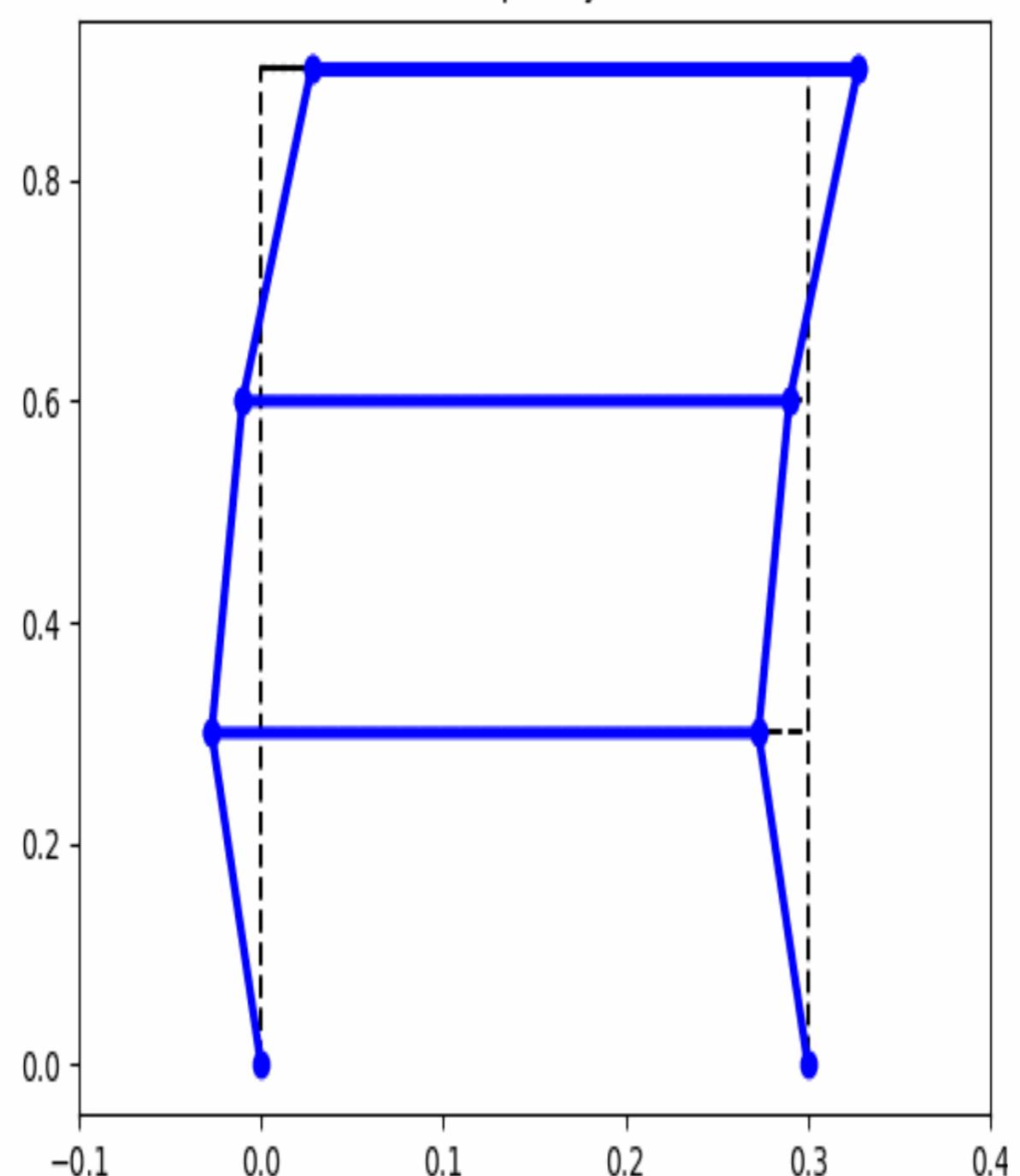


FDD

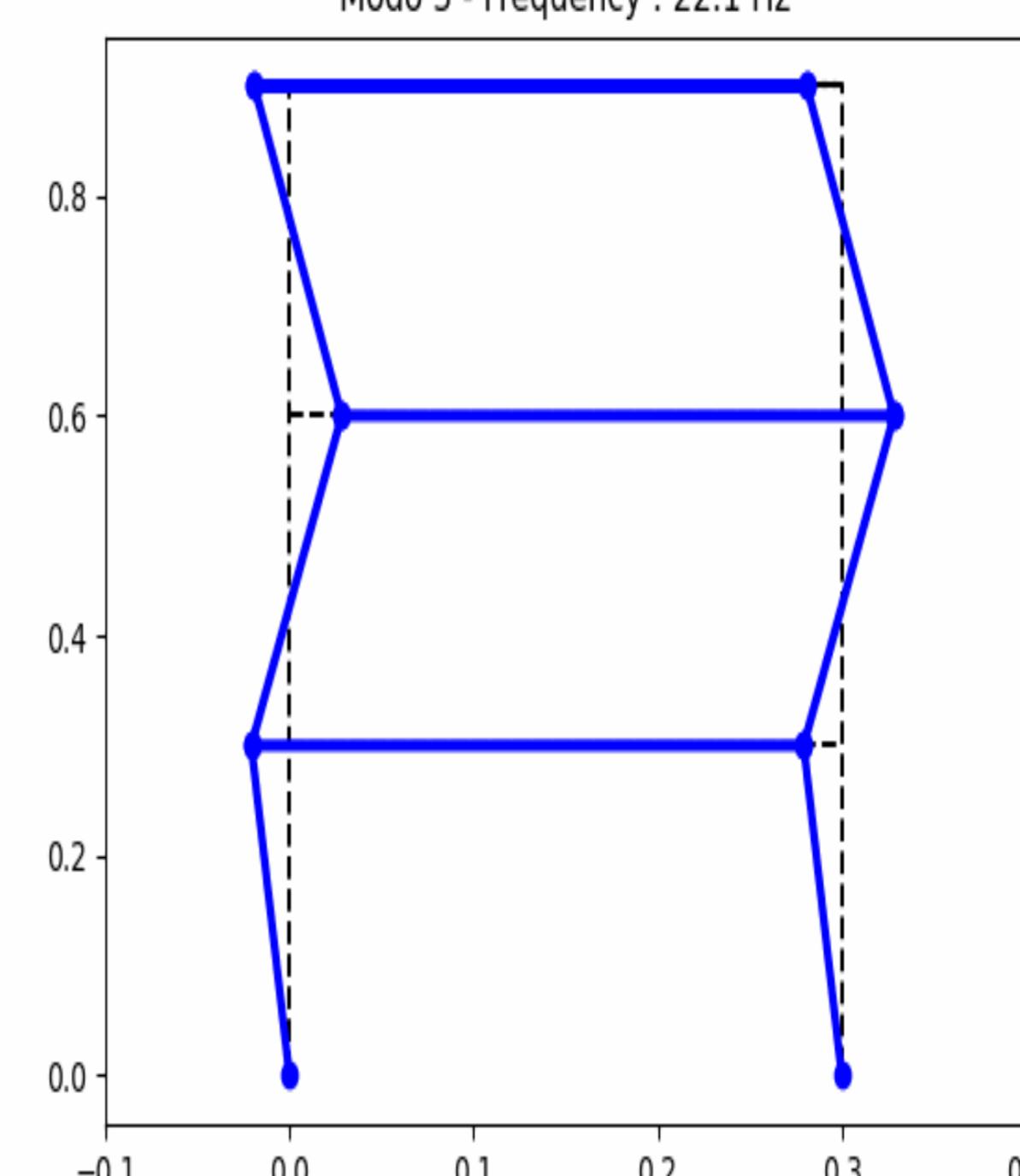
Modo 1 - Frequency : 4.61 Hz



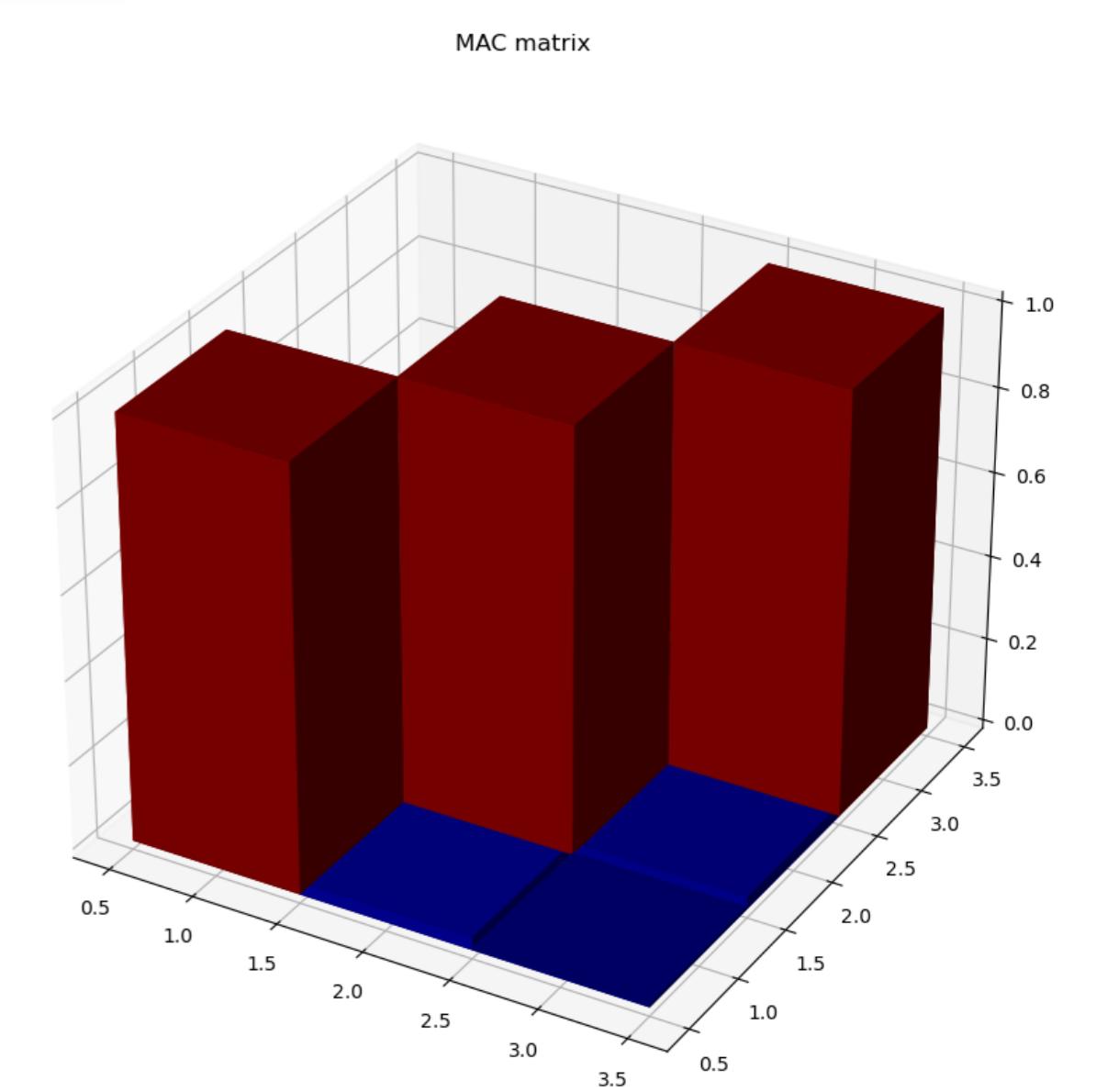
Modo 2 - Frequency : 13.67 Hz

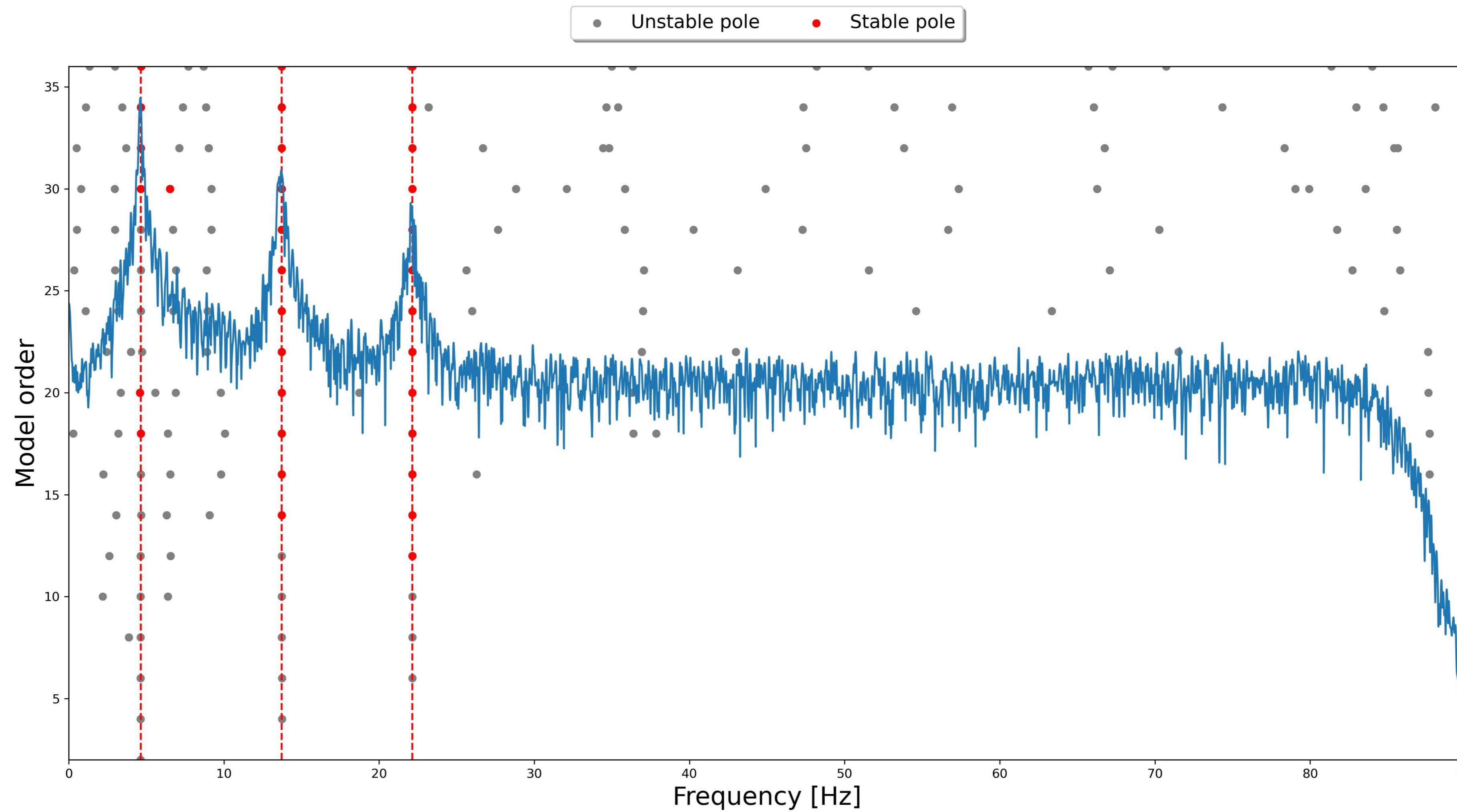


Modo 3 - Frequency : 22.1 Hz

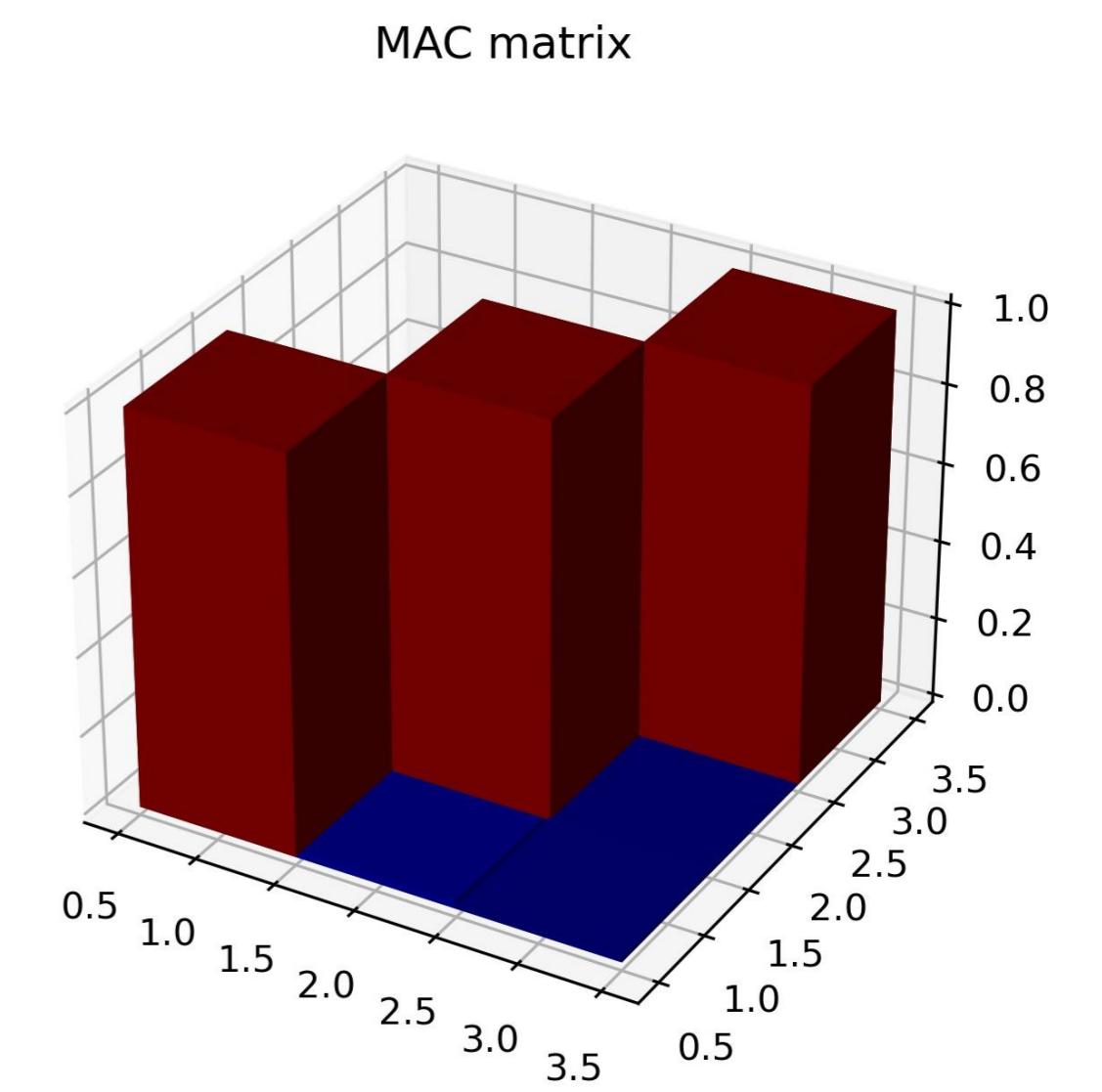
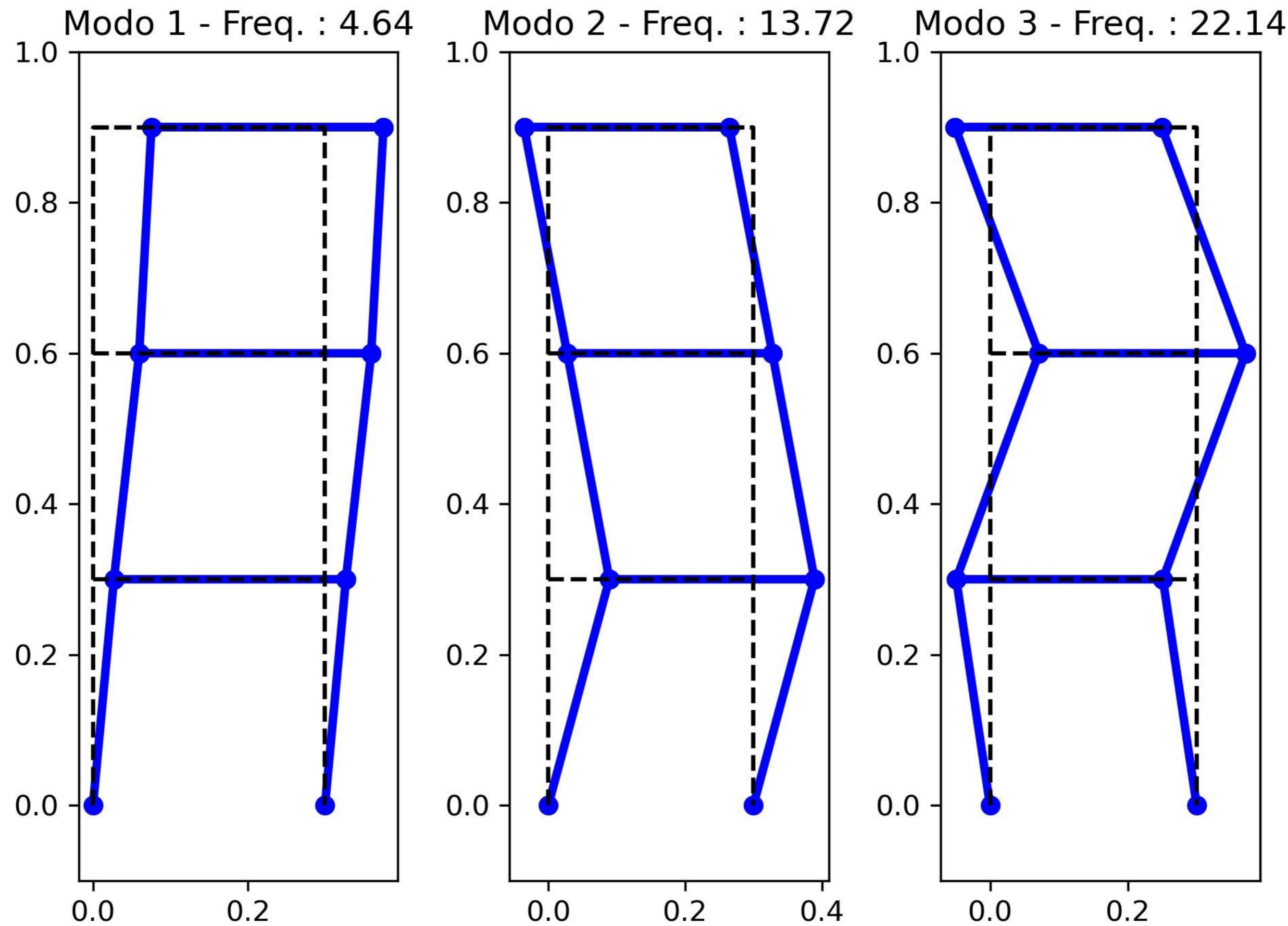


MAC matrix





COV-SSI



COMPARISON RESULTS

Mode N.	FDD		COV-SSI	
	f_i [Hz]	ξ_i [%]	f_i [Hz]	ξ_i [%]
1				
2				
3				



COV-SSI



UNIVERSIDAD
DE GRANADA