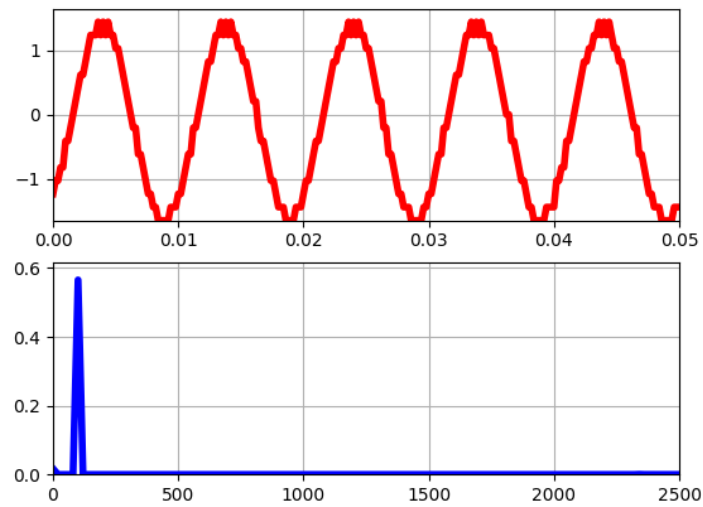
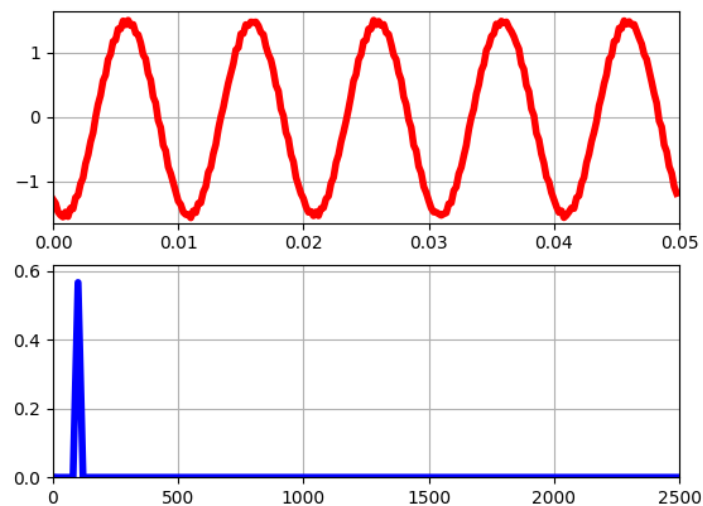


Digitalización a 4 bits de una señal de senoidal de 100 Hz



| Expression | Type | Value |
|---------------|-------|-------|
| $h_maxValue$ | q15_t | 7 |
| $h_minValue$ | q15_t | -8 |
| h_rms | q15_t | 0 |

Digitalización a 10 bits de una señal de senoidal de 100 Hz



| Expression | Type | Value |
|---------------|-------|-------|
| $h_maxValue$ | q15_t | 474 |
| $h_minValue$ | q15_t | -487 |
| h_rms | q15_t | 313 |

Código en main.c para el cálculo de valores máximo, mínimo y rms

```
arm_max_q15 ( adc, header.N, &h_maxValue, &h_maxIndex );  
arm_min_q15 ( adc, header.N, &h_minValue, &h_minIndex );  
arm_rms_q15 ( adc, header.N, &h_rms);
```

Códigos para digitación a 10 bits del ADC

Archivo main.c linea 202

```
adc[sample] = (int16_t) (medicion & 0x03FF) - MIDDLELEVEL_ADC;
```

Archivo tp1_testPrevUDPADC.py

```
adc[i]= ((int.from_bytes(msg[16+l[i]:16+l[i]+2], byteorder='little',  
signed=True))/512)*1.65
```

Códigos para digitación a 4 bits del ADC

Archivo main.c linea 203

```
adc[sample] = (int16_t) ((medicion & 0x03FF)>>6) - MIDDLELEVEL_ADC2;
```

Archivo tp1_testPrevUDPADC.py

```
adc[i]= ((int.from_bytes(msg[16+l[i]:16+l[i]+2], byteorder='little',  
signed=True))/8)*1.65
```

