

Sensor de nivel – Notas de Aplicación

Descripción general

Esta nota de aplicación describe como desarrollar un sensor de nivel de alta precisión utilizando el modelo uRAD Industrial. Este sensor de nivel puede servir para medir el llenado en tanques de materiales sólidos o líquidos, nivel de ríos o pantanos para aplicaciones de desbordamientos o cualquier aplicación donde sea necesario medir con precisión la distancia en la dirección recta.

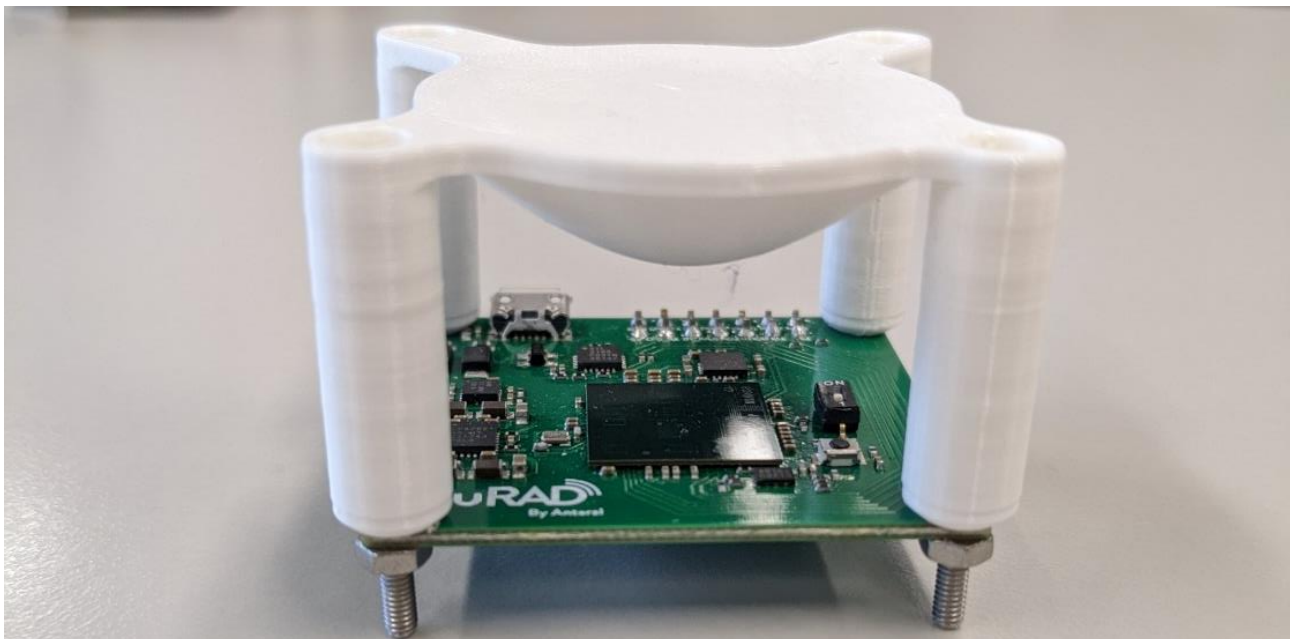
Dispositivos

Esta aplicación se puede desarrollar con nuestra solución uRAD Industrial. Se ha elegido este modelo de radar, principalmente por:

- Funciona en la banda de 60 – 64 GHz, una banda de libre emisión para aplicaciones industriales.
- Permite alcanzar precisiones de $\pm 1\text{mm}$.
- Resulta en un equipo muy compacto.
- Se puede conseguir un ángulo de visión muy estrecho de inferior a 10 grados añadiéndole una lente.

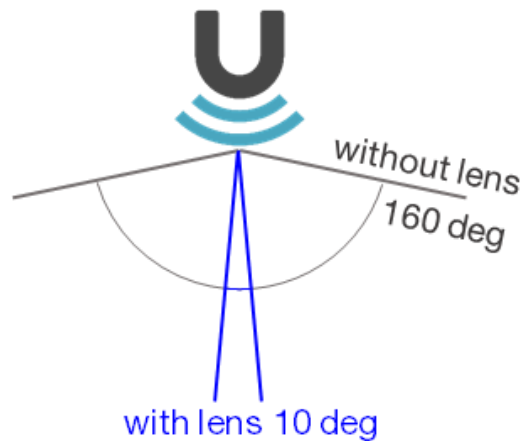
Montaje

El montaje básico es muy simple y consiste en uRAD Industrial + Lente de PLA. La lente viene integrada dentro de un soporte para que sea totalmente directo añadirla al radar.



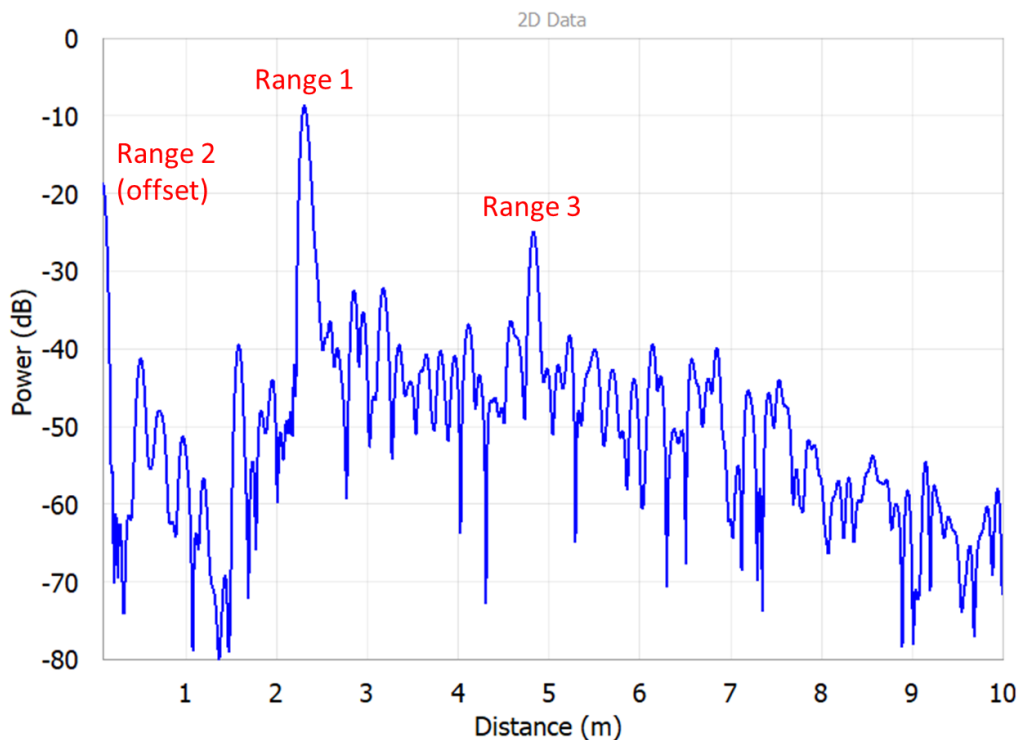
Con la lente se consigue reducir el ángulo de visión de uRAD Industrial de 160 grados a tan solo 10 grados. De esta manera, se enfoca toda la potencia emitida en la dirección recta.

Por ejemplo, para aplicaciones de llenado de tanques, esto es un aspecto crucial ya que el radar no debe ver las paredes del silo y que rebotes indeseados no le permitan medir con precisión.



Aspectos teóricos

El radar calcula las distancias a los objetivos mediante la transformada rápida de Fourier (FFT) de la señal recibida. Observando el espectro de la FFT, se pueden identificar los picos con más nivel, como las reflexiones de los objetos o superficies. El software de sensado de nivel permite visualizar este espectro y devuelve el rango de los tres picos con mayor nivel. A continuación, se muestra un ejemplo:



En el ejemplo, el radar se fija sobre una mesa apuntando al techo que se encuentra a una distancia de 2.5 metros. En la gráfica se identifican 3 picos.

- El más significativo, range 1, a 2.5 metros y que tiene una mayor amplitud es el techo, que proporciona la mayor reflexión.
- Hay un segundo pico, range 2, prácticamente en 0, que no corresponde a ningún valor real. A una distancia muy corta, generalmente inferior a 10 cm, siempre está presente un pico que no es real, llamado offset, que es inherente a la tecnología radar y que hay que descartar.
- Un tercer pico, range 3, a 5 metros corresponde con la doble reflexión producida por la mesa y el techo, que el radar es también capaz de detectar.

El software de sensado devuelve la distancia de los tres picos con más amplitud, pero la configuración permite seleccionar el rango de distancias donde buscar los tres picos. Por lo tanto, es posible eliminar fácilmente el offset de los resultados o descartar todos aquellos picos fuera del rango de interés.

Funcionamiento

Lo primero, se sube el firmware a uRAD Industrial. Hemos desarrollado un firmware específico para esta aplicación que se distribuye de manera gratuita con la compra cuando se solicita. El proceso de subir un firmware nuevo a uRAD Industrial está explicado en el manual de usuario general.

El funcionamiento del sensor es bastante simple. El dispositivo maestro, que controla uRAD, manda los comandos de configuración al radar por USB o UART. Una vez uRAD recibe los comandos, empieza a enviar los valores de medida.

La mayoría de los comandos son fijos por defecto. Solo hay que configurar unos pocos. Hemos hecho este proceso transparente para el usuario. Por lo tanto, se debe configurar:

```
#### CONFIGURATION PARAMETERS ####
maximum_distance = 44 # maximum distance to measure. From 11 to 44 meters
range_min = 0.2 # lower limit of the range of interest (min = 0 meters)
range_max = 44 # upper limit of the range of interest)
sampling_rate = 2 # min = 2, max = 20 samples per second
offset = 0 # apply an offset to all range measured values
```

- maximum_distance: el radar puede configurarse para medir una distancia máxima de 11 a 44 metros. Es mejor elegir el valor más bajo posible para tener la mejor resolución.
- range_min & range_max: rango de distancia de interés. De 0 metros hasta el valor de maximum_distance.
- sampling_rate: medidas por segundo que el radar envía al dispositivo maestro. De 2 a 20 muestras por segundo.
- offset: sumar una compensación (en metros) a los valores medidos.

Ejemplos de programación

Junto con el hardware, el firmware y los archivos de configuración, también proporcionamos diversos scripts de Python y C++ para controlar el radar, o bien por su conector USB o por el conector de pines a través de UART.

- level_sensing_UART.py

El radar se controla con el conector de pines usando una sola línea UART (mira el manual de usuario general para saber la conexión de pines), se usa el pin de reset y solo se envían los resultados de distancia (gráfico de la FFT y los valores de amplitud no se reciben). Los resultados se guardan en un archivo .txt si se selecciona.

```
#### OUTPUT RESULTS ####
saveResults = True
resultsFile_name = './output_files/results.txt'

#### UART PORT ####
port_name = '/dev/serial0'

#### RESET ####
reset_pin_number = 6    # GPIO pin number of the Master Device
reset = True
```

- level_sensing_USB.py

Similar al anterior pero controlado por USB. No incluye el pin de reset. **Por lo tanto, cada vez que quieras lanzar el código otra vez, se debe resetear manualmente el radar con el botón físico de reset.**

```
#### OUTPUT RESULTS ####
saveResults = True
resultsFile_name = './output_files/results.txt'

#### USB PORT ####
configPort_name = 'COM2'
dataPort_name = 'COM1'
```

- level_sensing_USB_GUI.py:

El más completo porque permite obtener resultados de distancia, pero también la amplitud de los picos. Los comandos de configuración incluyen mandar los valores de IQ para el gráfico FFT. Por lo tanto, el gráfico de picos (FFT) se puede visualizar si se selecciona.

```
#### OUTPUT RESULTS ####
saveResults = True
plotSpectrum = True
calculateAmplitude = True
resultsFile_name = './output_files/results.txt'
```

```
#### USB PORT ####
configPort_name = 'COM2'
dataPort_name = 'COM1'
```

- level_sensing_function.py

Este ejemplo es un script que muestra como crear una función simple para tomar una sola muestra cuando se llama a la función. Por lo tanto, permite al usuario tomar solo una muestra en el momento deseado.

Comandos de configuración

Como se ha comentado anteriormente, el radar necesita recibir algunos comandos para ser configurado. En los scripts de ejemplo esto es transparente para el usuario. A continuación, se muestran estos comandos de configuración, donde se resaltan los parámetros que se configura:

```
flushCfg
dfeDataOutputMode 1
channelCfg 1 1 0
adcCfg 2 1
adcbufCfg 0 1 1 1
profileCfg 0 60 7 7 114.4 0 0 33.71 1 512 5000 0 0 48
chirpCfg 0 0 0 0 0 0 0 1
frameCfg 0 0 10 0 500 1 0
lowPower 0 0
guiMonitor 1 0 0 0 0 1
RangeLimitCfg 2 1 0.3 10.0
sensorStart
```

- Pendiente del chirp / Distancia máxima

Con el parámetro resaltado **33.71** en la línea de **profileCfg** se selecciona la pendiente del chirp, que a su vez define la distancia máxima a medir. La distancia máxima puede ser de 11 a 44 metros porque el chirp slope puede variar de 33.71 a 8.4275. Ambas magnitudes se relacionan con la siguiente fórmula:

$$\text{pendiente del chirp} = 11 * \frac{33.71}{\text{distancia máxima}}$$

- Refresco

Con el parámetro resaltado **500** de **frameCfg** se puede seleccionar cada cuantos milisegundos se envía una muestra. El refresco más rápido son 50 milisegundos, que corresponde con 20 muestras por segundo.

- Enviar resultados de IQ / Gráfico FFT

Con el parámetro resaltado en la línea **guiMonitor**, se puede seleccionar si deseas que el radar mande los valores en brutos de IQ para luego pintar la gráfica de la FFT junto con los valores de resultados (1) o si por el contrario no deseas los valores de IQ y que se envíen solo los valores de resultados finales (0). Proporcionamos dos scripts, uno donde se dibuja la gráfica y se guardan los resultados, y otro script donde solo se guardan los resultados.

- Rango de distancias

Con los dos parámetros resaltados 0.3 y 10.0 del parámetro **RangeLimitCfg** se define la distancia mínima (0.3 metros en el ejemplo) y la distancia máxima (10 metros en el ejemplo) donde te interesa medir para descartar cualquier otro pico fuera de ese rango.

Medidas reales

A continuación, se presenta un caso práctico real.

La configuración es la siguiente. El radar se coloca a una distancia de 1.7 metros de una chapa metálica. A mitad de distancia, se colocará un cristal. Se van a medir dos situaciones, con el cristal perpendicular al radar y con el cristal inclinado.



El radar está conectado a un ordenador. Se alimenta directamente por el USB por donde también se configura y recibe los datos.

El script que se utiliza es *level_sensing_USB_GUI.py* ya que queremos mostrar la gráfica de pico además de guardar los resultados.

La configuración es:

```
#### CONFIGURATION PARAMETERS ####
maximum_distance = 11 # maximum distance to measure. From 11 to 44 meters
range_min = 0.3 # lower limit for the range of interest (min = 0 meters)
range_max = 3 # upper limit for the range of interest
sampling_rate = 2 # min = 2, max = 20 samples per second
offset = 0 # apply an offset to all range measured values
```

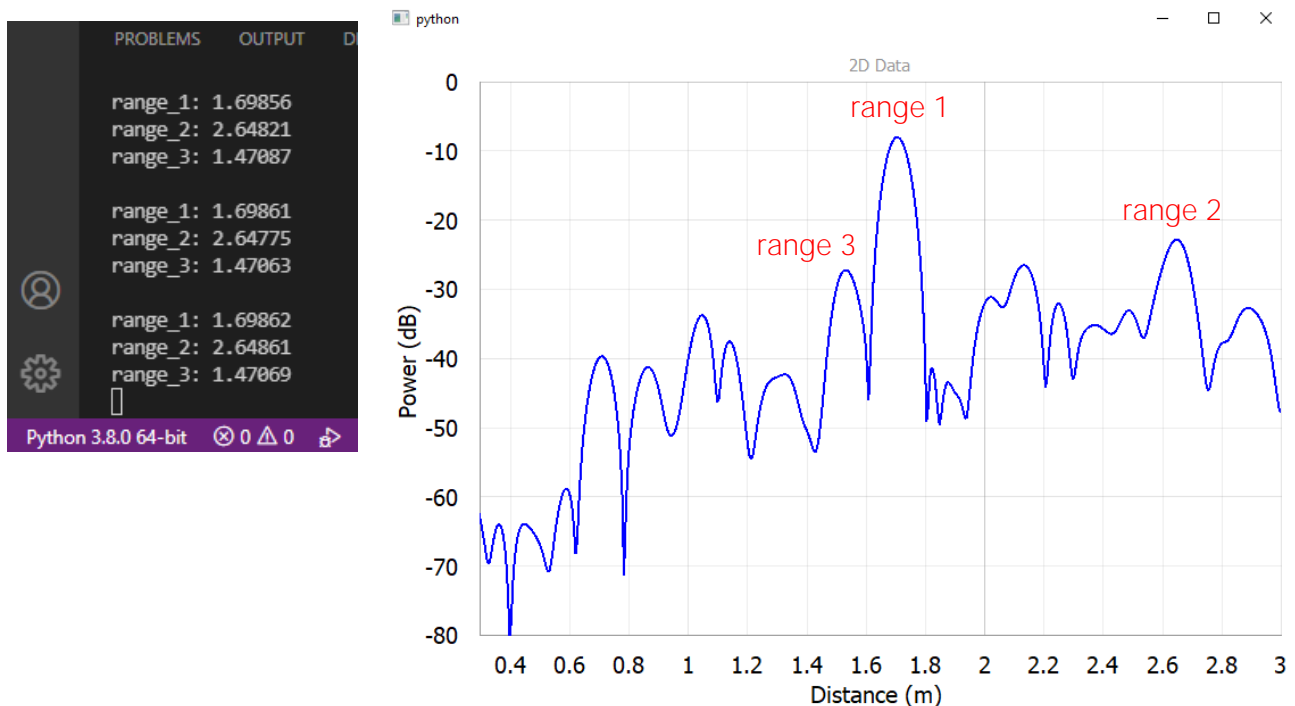
Conectamos el radar y vemos los puertos COM que se han asignado. Configuramos el resto de los parámetros del script como sigue:

```
#### OUTPUT RESULTS ####
saveResults = True
plotSpectrum = True
calculateAmplitude = False
resultsFile_name = './output_files/results.txt'

#### USB PORT ####
configPort_name = 'COM2'
dataPort_name = 'COM1'
```

- Sin cristal

Tras lanzar el script, obtenemos los siguientes resultados medidos:

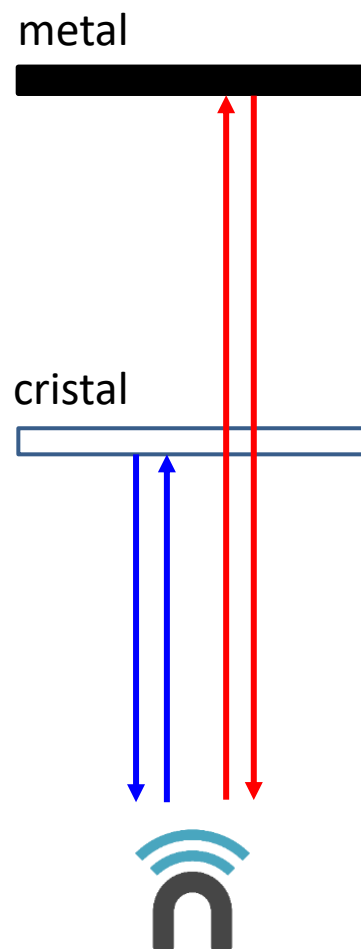


Observando los resultados tanto en la consola de Python como en la gráfica, vemos los tres picos que se identifican. Los resultados se ordenan de mayor a menor amplitud del pico.

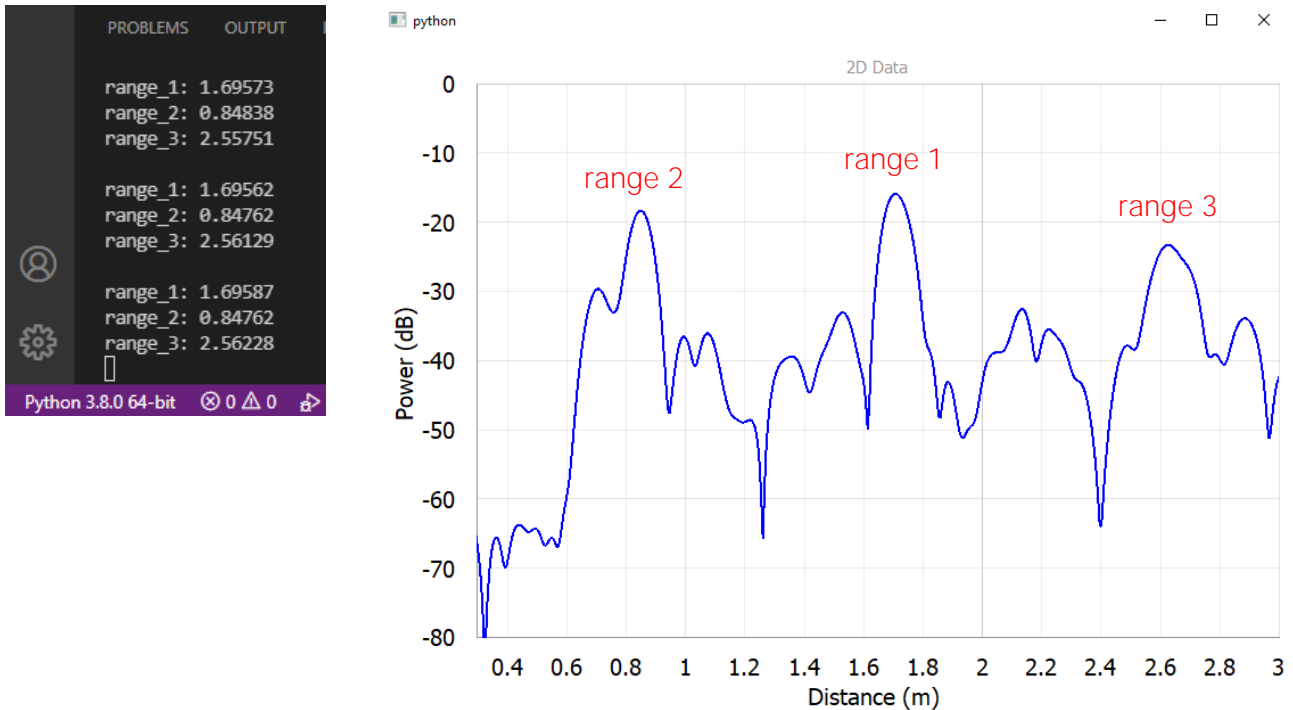
A la distancia de 1.69856 metros se ha medido el cristal con una amplitud muy por encima de los otros dos picos, que en realidad no se corresponden con ninguna superficie de interés.

El resto de picos del espectro, incluido range 2 y 3, se producen por el propio procesado de la FFT y de rebotes del entorno.

- Cristal colocado perpendicularmente



Los resultados obtenidos son:

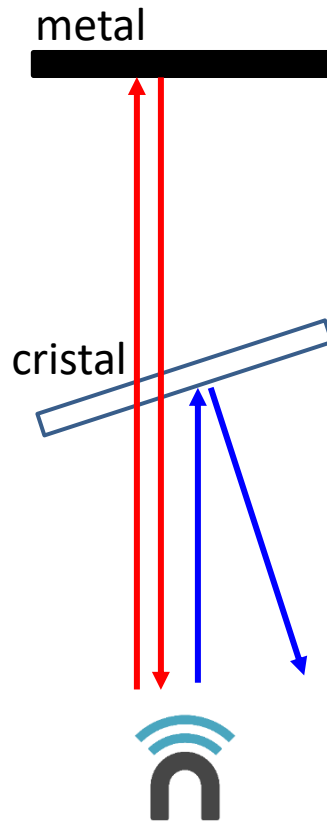


En este caso se observa como aparece el cristal a una distancia de 0.84838 metros y el metal sigue apareciendo a una distancia de 1.69573 metros. En este caso, el nivel de amplitud de ambos picos es muy similar, por lo que el valor de range 1 podría estar saltando entre ambos valores o incluso atribuir el pico de mayor amplitud al cristal. Además, se observa el pico correspondiente al metal ha bajado de amplitud como consecuencia de la atenuación de la onda a través del cristal.

- Cristal colocado con ángulo

En este caso, inclinamos el cristal para observar cómo su pico desaparece. Con una inclinación de 20-30 grados es suficiente para que el pico desaparezca.

Esto puede ser muy útil, por ejemplo, en situaciones reales donde se requiera aislar con una tapa el radar o donde es inevitable atravesar varios materiales. En ese caso, una inclinación en esa superficie mitigará el efecto del rebote.



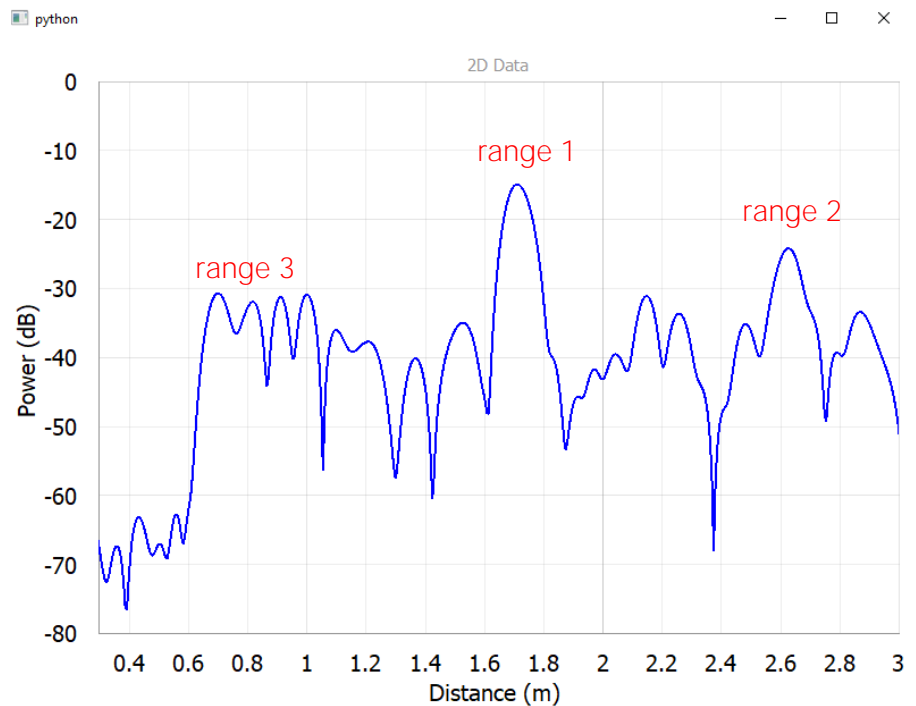
Al inclinar el cristal, la mayor parte de la potencia reflejada no incide sobre el radar, como se muestra en la imagen anterior.

```

PROBLEMS OUTPUT
range_1: 1.69778
range_2: 2.63228
range_3: 1.00076

range_1: 1.69745
range_2: 2.63226
range_3: 1.00006

range_1: 1.69750
range_2: 2.62830
range_3: 0.69223
    
```



Se observa como el pico del cristal se ha mitigado completamente. En esta nueva situación, range 1 se identifica inequívocamente con el metal, siendo su amplitud muy superior al resto de picos.

El archivo de texto generado con los resultados, en este caso y tal como se indica en la configuración, se llama *results.txt* y se guarda en la carpeta *output files*. El formato es el siguiente:

```
1.69778 2.63228 1.00076 1622787713.261
1.69745 2.63226 1.00006 1622787713.762
1.69750 2.62830 0.69223 1622787714.261
1.69761 2.63123 0.69210 1622787714.762
1.69776 2.63211 1.00037 1622787715.260
```

Se guarda en cada fila los valores de cada medida, un frame de cada media, ordenando por columnas:

```
range1    range2    range3    time_stamp
```