

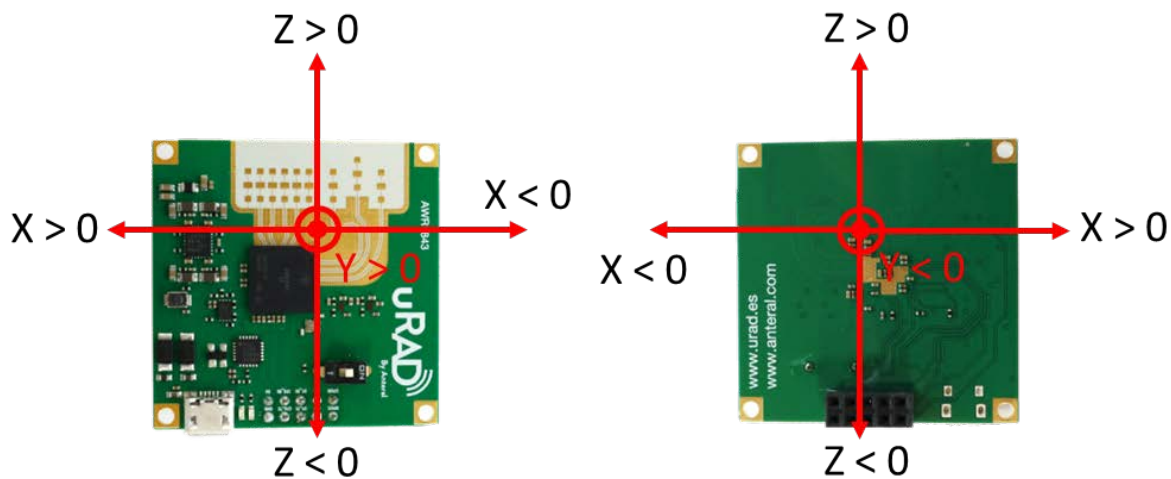
Radar de Corto Alcance – Notas de Aplicación

Resumen

El firmware y el software del radar de corto alcance demuestran la capacidad de uRAD Automotive HPA para funcionar como un radar de corto alcance para detectar obstáculos hasta 10 metros y un campo de visión de 140 grados. Los obstáculos se posicionan en el espacio 2D (rango y azimut) donde se omite la información de altura.

Posición del radar

Los ejes del radar se definen de acuerdo a la siguiente imagen.



Ten en cuenta que **la nube de puntos no contiene información del eje Z** porque los obstáculos se posicionan solo en el plano XY.

Por lo tanto, la orientación del radar debe ser:

- Eje X paralelo al suelo (± 70 grados campo de visión en azimut)
- Eje Y apuntando al frente



Firmware y configuración chirp

El firmware utilizado es el Out of Box Demo, el firmware normal para obtener la nube de puntos en función de una configuración de chirp deseada. La configuración de chirp se ha elegido para maximizar el alcance y la precisión del ángulo en distancias cortas.

Parámetro	Valor	Detalles
Rango máximo	10 m	Distancia máxima a la que el radar puede detectar un objeto.
Resolución en Rango	4 cm	Capacidad para distinguir entre dos o más objetivos en el mismo rumbo pero en diferentes rangos.
Velocidad máxima	4.7 m/s	Velocidad máxima obtenida
Resolución en velocidad	0.3 m/s	Capacidad para distinguir entre dos o más objetivos en el mismo rango pero moviéndose con diferentes velocidades.
Precisión en ángulo	1 grado	Precisión para posicionar en ángulo el objeto detectado.
Muestras por segundo	30 fps	Número de detecciones por segundo

El radar envía la nube de puntos. La nube de puntos contiene la información de (X,Y,Z,velocidad, SNR,ruido) de los puntos detectados. Cada objeto detectado genera uno o más puntos en la nube de puntos. El número de puntos que genera cada objeto depende de muchos factores (posición, velocidad, forma, materiales, etc.).

- Aunque la velocidad máxima es de 4,7 m/s = 17 km/h, el radar puede detectar los objetos que se mueven a mucha más velocidad debido a un modo de velocidad extendida.
- El valor de Z siempre es 0 porque la configuración chirp se selecciona para maximizar la precisión en el plano 2D.

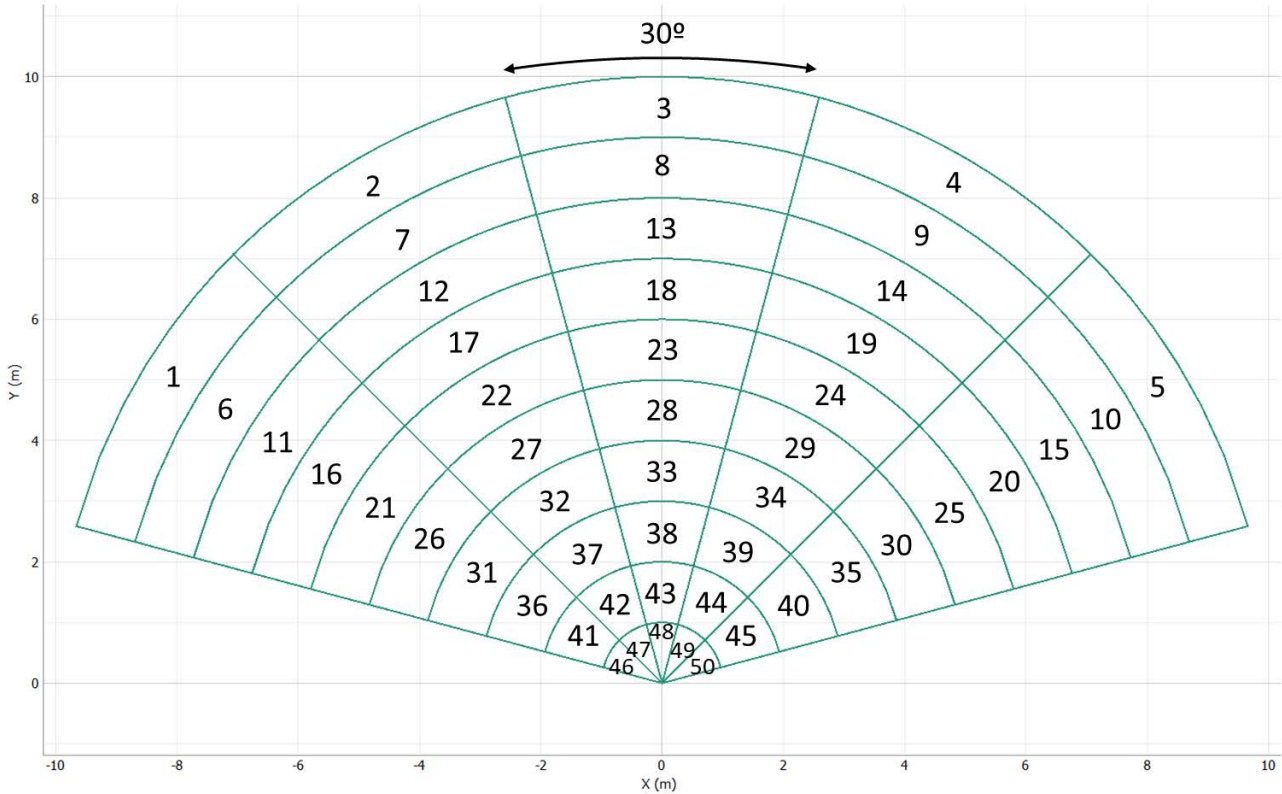
Detección

El punto clave de esta aplicación es que el área frontal del radar se divide en **50 zonas de detección**. Estas zonas de detección tienen un tamaño de **1 metro por 30 grados**.

Si se detectan uno o más puntos dentro de cada zona durante más de 3 muestras consecutivas, la zona está activa y podemos concluir que hay un objeto ocupando esa zona.

Esta condición de más de 3 muestras consecutivas se implementa para reducir los falsos positivos causados por el clutter.

En los resultados generados, las zonas se guardan como una matriz de 10 filas y 5 columnas. Cada posición de la matriz se corresponde con una zona, según la siguiente figura.



Si hay un objeto en la zona, la posición correspondiente en la matriz se establece en 1, si NO hay objeto, el valor de la posición es 0.

Matriz de zonas detectadas

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25
26	27	28	29	30
31	32	33	34	35
36	37	38	39	40
41	42	43	44	45
46	47	48	49	50

Software

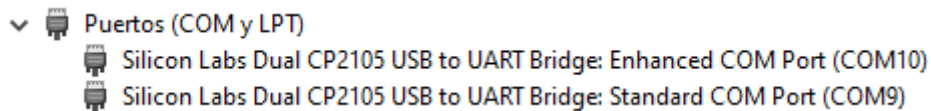
Se han programado dos scripts de python para trabajar con el radar. Estos scripts envían la configuración chirp al radar, reciben la nube de puntos y crean la matriz de zonas detectadas. Tanto la nube de puntos como la matriz se pueden guardar en archivos .txt.

- **short_range_radar_USB.py**

Este script es para usar uRAD Industrial conectado por USB. En el script, se deben configurar los puertos COM donde se conecta el radar..

```
configPort_name = 'COM2'  
dataPort_name = 'COM1'
```

Los puertos COM se identifican fácilmente en el *Administrador de Dispositivos*:



- Enhanced port es el puerto de configuración
- Standard port es el puerto de datos

- **short_range_radar_single_UART.py**

Este script es para usar uRAD Automotive HPA mediante el conector pin usando un solo canal UART. Por ejemplo, este script es útil cuando se usa uRAD junto con el adaptador de PCB para Raspberry Pi. El nombre del puerto tiene que ser introducido. Por ejemplo, en la Raspberry Pi es:

```
Port_name = '/dev/serial0'
```

En este script también está disponible la posibilidad de hacer un reinicio usando el pin Reset del conector pin.

```
reset = True
```

Se debe configurar el nombre/número de pin del dispositivo maestro donde se conecta el pin de reinicio de uRAD. Usando el adaptador de PCB, este pin es GPIO número 5.

```
PinReset = OutputDevice(5)
```

Se pueden configurar dos parámetros para limitar el rango y el ángulo de azimut. De esta manera, la nube de puntos se limita y el usuario puede enfocarse en buscar objetos en la dirección recta o en un rango más corto.

```
### CONFIGURATION PARAMETERS ###  
max_azimuth = 90 # View angle of the radar. Possible values are 30, 90 or 150  
max_range = 8 # Detection max range of the radar. Possible values are 4, 6, 8 or 10
```

Solo se permiten algunos valores para el rango y el azimut:

```
possible_ranges = [4,6,8,10]  
possible_azimuth = [30,90,150]  
if max_range not in possible_ranges or max_azimuth not in possible_azimuth:  
    sys.exit("Range or value not set corretly")
```

Resultados

La nube de puntos y la información de la zona se guardan en dos .txt diferentes. si la variable correspondiente está establecida en True.

```
savePointCloud = True  
saveDetectionZones = True
```

Y sus nombres.

```
pointCloud_fileName = './output_files/PointCloud.txt'  
detectedZones_fileName = './output_files/DetectedZones.txt'
```

- **PointCloud.txt**

Cada línea de este archivo contiene la información de la nube de puntos en cada muestra. Se guarda consecutivamente en cada columna

[X (m)] [Y (m)] [Z (m)] [Velocity (m/s)] [SNR] [Noise]

de cada punto detectado de la nube de puntos. Al final, se incluye la marca de tiempo, como la cantidad de segundos desde el 1 de enero, 1970, 00:00:00 (UTC).

- **DetectedZones.txt**

Cada línea de este archivo contiene la información de la matriz completa en cada muestra. Se guarda consecutivamente en cada columna:

[1] [2] [3] ... [50] → cada posición guarda 0 (NO obstáculo), 1 (SI obstáculo)

Al final, se incluye la marca de tiempo, como la cantidad de segundos desde el 1 de enero, 1970, 00:00:00 (UTC).

Visualizador

También se ha programado un visualizador para facilitar la visualización de los resultados en tiempo real al utilizar el radar por USB.

La visualización se puede encontrar en la carpeta *GUI*. Ejecute el script de python **short_range_radar_GUI.py** para usarlo.

Con este visualizador:

1. Limitar los resultados en rango y ángulo acimutal según los parámetros de configuración.
2. Guardar el archivo .txt de la nube de puntos (no se puede guardar la matriz de zonas)
3. Reproducir resultados previamente guardados.

En este visualizador, la nube de puntos se dibuja como puntos negros y las zonas se colorean cuando están activas y, por lo tanto, hay un obstáculo en ellas.

