

# MANUALE INSTALLAZIONE SENSORE **MINIWE MS72SF1**

## Componenti :

- 1) Radar MS72SF1
- 2) USB to TTL per configurazione porta seriale
- 3) Estensione cavo USB
- 4) Cavo a 5 conduttori pin

## Collegamento PIN a USB TTL :

Connettere i PIN (con su scritto il nome) alle 5 componenti in ordine...come possiamo notare nella *Figura 1*, con una differenza importante : connettere il pin TX a RX (dell USB) e il pin RX a TX (dell USB). Questa connessione incrociata è necessaria perché il dispositivo che trasmette dati deve essere collegato al dispositivo che li riceve, e viceversa. Ovviamente il radar è connesso tramite seriale(USB TTL) al PC. Il dato trasmesso da un dispositivo verrà ricevuto correttamente dal dispositivo opposto. Senza questa connessione incrociata, i dati inviati da un dispositivo non potrebbero essere ricevuti dall'altro, impedendo qualsiasi comunicazione.

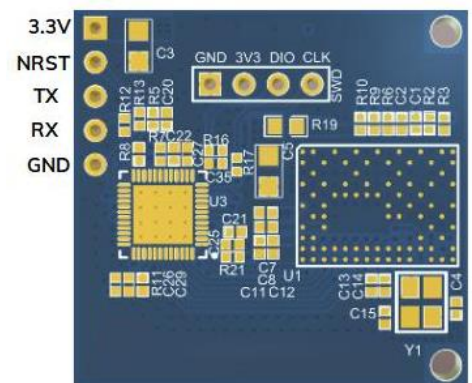


Figura 1 – Sensore MS72SF1



Figura 2

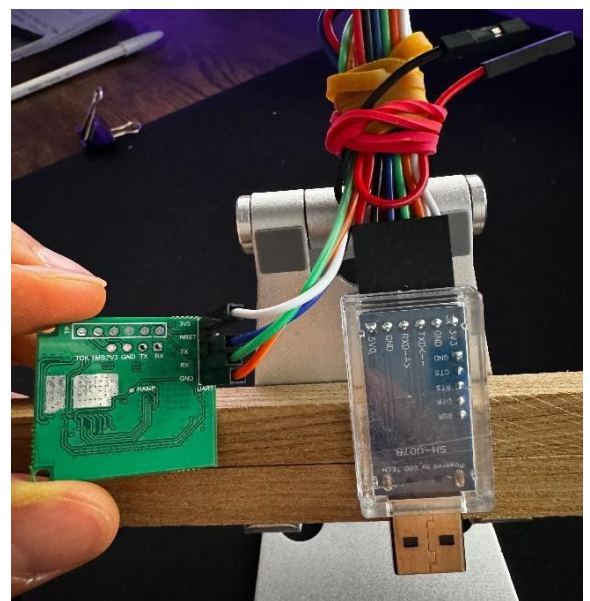


Figura 3

Il radar supporta due possibili configurazioni "CeilMount" ovvero montato al soffitto, oppure "SideMount" ovvero montato alla parete. Prima di montarlo bisogna installare il firmware prescelto per la configurazione all' interno del radar tramite l' applicazione "FirmwareUpdateApp.exe" e seguire i passaggi :



Figura 4 – Applicazione eseguibile per installazione sensore

Una volta connessa al PC la pennetta USB TTL tramite l' estensione USB, il pc dovrebbe riconoscere la connessione ad una nuova porta seriale da parte del radar, con il bottone **1** si fa il refresh di tutte le porte e dovrebbe apparirne una nuova, **2** la seleziono, **3** apro la comunicazione fra il radar e il PC , nel passaggio **4** bisogna selezionare un file .Bin a scelta fra le due configurazioni, quindi bisognerà selezionare o il file "CeilMount.bin" oppure "SideMount.bin" così che venga installato nel sensore con il bottone in posizione **5**.

Una volta effettuato il settaggio e montato il radar a parete o sul soffitto, dobbiamo prendere dimestichezza con il software (**SSCOM5**) che gestisce lo scambio di informazioni fra il radar e il computer.

SSCOM5 è un software di terminale seriale usato per la comunicazione con dispositivi via porta seriale (come nel nostro caso), come radar o altri strumenti che richiedono configurazione.

Questa è la sua interfaccia :

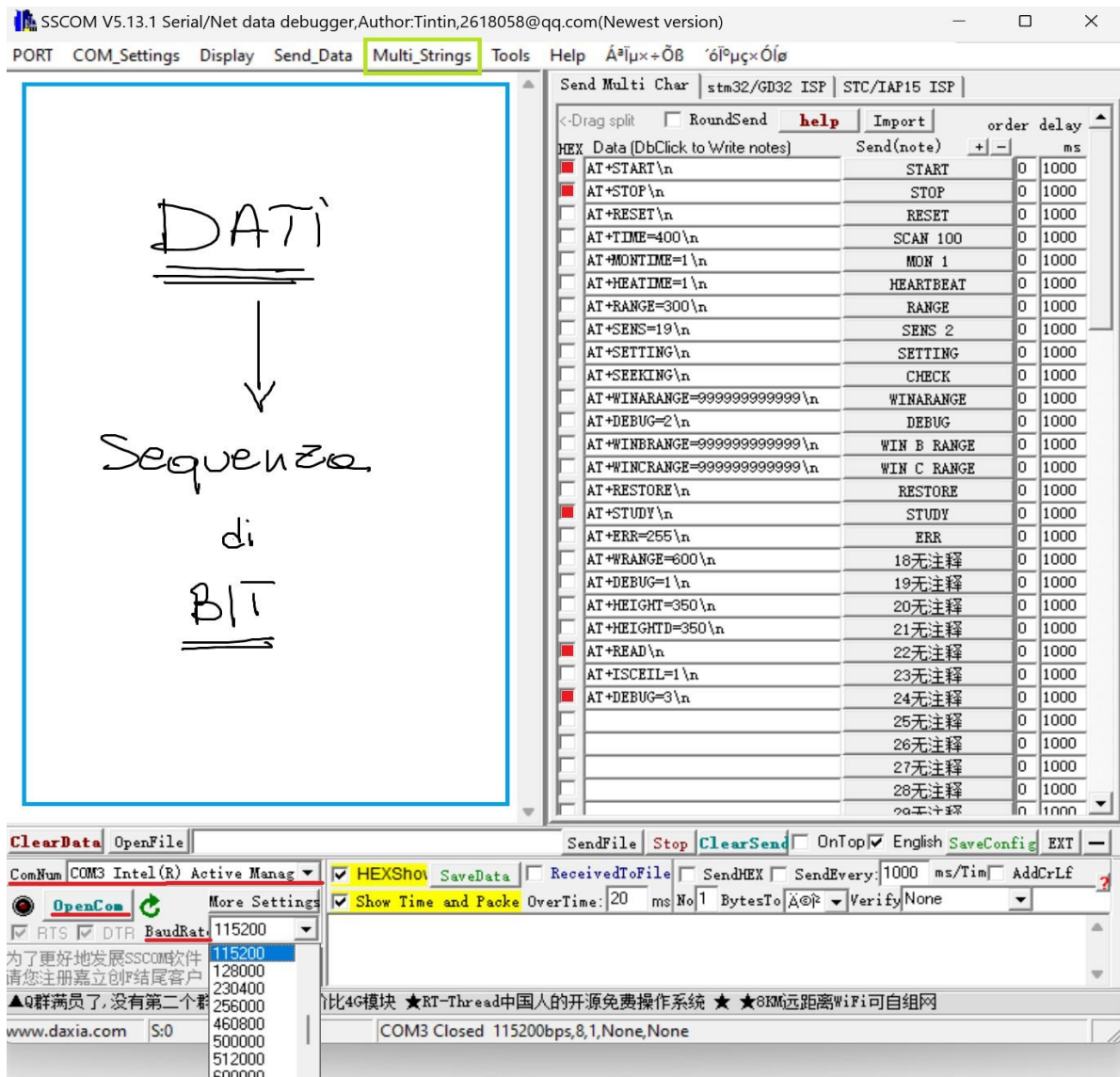


Figura 5 – SSCOM

I primi passaggi sono i seguenti :

- 1) Selezionare la porta seriale esatta in "ComNum" e aprire la connessione con il pulsante "OpenCom" .
- 2) Impostare il BaudRate a seconda del sensore, nel nostro caso 115200.
- 3) Selezionando in alto "Multi\_Strings" si apre la sezione sulla destra dove possiamo passare velocemente le istruzioni al sensore invece di scriverle a mano. Come scritto nel manuale bisogna configurare il radar una volta montato con varie istruzioni (vedi millimeter-wave radar development guidance).

I comandi per una giusta configurazione sono i seguenti :

- **AT+RESTORE**\n -> per ripristinare la configurazione da zero.
  - **AT+TIME=400**\n -> frequenza di invio di impulsi in ms.
  - **AT+STUDY**\n -> inizio a "studiare" l' ambiente circostante.
  - **AT+HEIGHT=300**\n , **AT+HEIGHTD=300**\n ... -> varie impostazioni (Altezza sensore, posizione in cm, distanza radiale del sensore e altre scritte nel manuale)
  - **AT+DEBUG=3**\n -> protocollo modalità operativa per il tracking delle persone.
  - **AT+START**\n -> invio e ricezione dei dati da parte del radar.
- 

Altri comandi importanti sono :

- **AT+STOP**\n -> stop del funzionamento del radar.
- **AT+READ**\n -> per leggere la configurazione di come è stato impostato.

**4)** Nel riquadro blu verranno visualizzati i dati, ovviamente come sequenze di bit che andranno interpretate per capire i valori delle varie coordinate. Per ottenere una giusta visualizzazione , bisogna selezionare HEXShow in giallo e Show Time se vogliamo ottenere anche l' orario di ricezione dei punti.

E' consigliato fare la procedura di configurazione ogni volta che si usa il radar.

### **Problemi riscontrati** **\*\*IMPORTANTE\*\*** :

Il radar connesso a un computer portatile inizialmente dopo qualche secondo dallo START si disconnette dalla porta USB... questo avviene quando il dispositivo cerca di consumare un quantitativo maggiore di energia, superando così la potenza nominale della porta, che di standard è 0,5 amp. Per risolvere questo problema ho eseguito i seguenti passaggi :

**1)** Modificato le impostazioni di gestione dell' alimentazione delle porte USB :

- Aprire gestione dispositivi (WIN+R) , scrivere *devmgmt.msc* nella finestra di dialogo e poi clicca su OK.
- Fai doppio click su "Universal serial Bus controller" e quindi sul primo elemento dell'hub principale USB.
- Seleziona la casella accanto a "consenti al computer di spegnere il dispositivo per risparmiare energia". Clicca su OK. Fallo per ogni USB.

2) Ridurre la frequenza di invio con il comando AT+TIME=400\n, di default è 1000ms questo ci consente di ricevere dati senza disconnessione.

Infatti la disconnessione improvvisa avviene solo da portatile quando si va ad "alta frequenza" , mentre se si opera da una workstation/ computer fisso, il problema non si pone.

## Elaborazione dei dati – PERSONE (DEBUG = 2/3 )

SSCOM è in grado di generare un file .TXT dove vengono salvati i dati che riguardano coordinate, sia di persone, che nuvole di punti individuate dal radar. I dati sono rappresentati in bytes, non sono di facile interpretazione, quindi ho creato uno script che manipola il file generato per ottenere una rappresentazione delle coordinate in float, seguendo come linea guida la documentazione ufficiale per le varie conversioni :

Field		Number of bytes	Description
HEAD		8	Frame header, fixed\x01\x02\x03\x04\x05\x06\x07\x08
LENGTH		4	Whole frame data length (uint32)
FRAME		4	Frame number (uint32)
TLVs		4	TLVs=1 followed by point cloud information (uint32)
POINTLENTH		4	Point cloud length is always 0 (uint32)
TLVs		4	TLVs=2 followed by person information (uint32)
TRACKLENTH		4	Length of person data (number of persons = TRACKLENTH/32) (uint32)
Personnel 1	ID	4	Personnel markers (uint32)
	Q	4	reserve(uint32)
	X	4	X/Y/Z coordinates of the person and the speed (float)
	Y	4	
	Z	4	
	Vx	4	
	Vy	4	
	Vz	4	
.....			
Personnel n	ID	4	Personnel markers (uint32)
	Q	4	reserve(uint32)
	X	4	X/Y/Z coordinates of the person and the velocity (float), in units: coordinates in m and velocity in m/s, to two decimal places. Single precision floating point type according to the standard for binary floating point arithmetic (IEEE 754),with the small endunwrappedbefore. <a href="https://www.binaryconvert.com/convert_float.html">https://www.binaryconvert.com/convert_float.html</a> <a href="https://www.cnblogs.com/guanshan/articles/guan022.html">https://www.cnblogs.com/guanshan/articles/guan022.html</a>
	Y	4	
	Z	4	
	Vx	4	
	Vy	4	
	Vz	4	

Figura 6 – Parametri e descrizione DEBUG 3

Il file di testo che genera SSCOM in modalità DEBUG 3 è così (esempio 1 riga) :

```
[10:04:56.085]IN;û;ô01 02 03 04 05 06 07 08 80 00 00 00 8A 00 00 00 01 00 00 00 00 00 00  
00 02 00 00 00 60 00 00 00 01 01 01 00 00 00 00 00 59 9F 7F 3F FD 9E B9 BF DD F1 11 40  
00 00 00 00 00 00 00 00 00 00 00 00 01 01 00 00 01 00 00 00 6B AE 9E BB CF DD E2 3E 54  
DC E6 3F 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 02 00 00 00 0C 68 1E BF D2 5A  
B0 BF 14 6B AC 3F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

Lo script Python che ho realizzato esegue varie fasi per convertire e elaborare dati esadecimali da un file di testo in un file CSV finale, ma soprattutto è utile per poi studiare/elaborare i dati in modo semplice e preciso . Il flusso di lavoro dello script può essere suddiviso come segue:

### 1. Lettura e conversione del file di testo :

- Lo script legge il file di testo contenente dati esadecimali strutturati generato da SSCOM.
- Estrae i timestamp e i dati esadecimali dalle righe del file.
- Divide i dati esadecimali in blocchi di dimensioni specifiche per una migliore organizzazione.
- Scrive questi dati organizzati in un file CSV intermedio, aggiungendo un'intestazione con nomi di colonne appropriati (*Figura 6*) .

### 2. Elaborazione del file CSV intermedio :

- Lo script legge il file CSV intermedio creato nella prima fase.
- Elimina le colonne che contengono solo valori nulli per pulire il dataset.
- Rimuove colonne specifiche non necessarie per l'analisi finale.
- Filtra le righe per rimuovere quelle senza valori essenziali.
- Applica diverse funzioni di trasformazione ai dati rimanenti:
  - **Numero di persone:** Converte i dati esadecimali in un numero che rappresenta il numero di persone.
  - **Coordinate e velocità:** Converte valori esadecimali in numeri in virgola mobile (float) con due cifre decimali.
  - **Dati binari:** Converte stringhe binarie in numeri decimali.
  - Scrive i dati trasformati in un file CSV finale.



### 3. Automazione tramite riga di comando :

- Lo script è progettato per essere eseguito dalla riga di comando, accettando il percorso del file di testo di input come argomento.
- Esegue automaticamente le due fasi di conversione e elaborazione per generare il file CSV finale.

### Utilizzo dello Script

Per utilizzare lo script, digitare dalla riga di comando:

```
> python pointsPerson.py <path/to/input_file.TXT> <d2/d3>
```

Questo comando legge il file di testo specificato, esegue le operazioni di conversione e elaborazione descritte, e genera un file output\_processed.csv contenente i dati finali trasformati. Attenzione lo script può essere utilizzato anche se in input abbiamo dei dati relativi alla modalità DEBUG 2 (il file è strutturato diversamente) basterà dare al comando "d2" come secondo parametro per ottenere un risultato esatto. Ovviamente otteniamo sempre e comunque dati relativi a persone.

### Risultato :

Time	TLV2	NumPeople	ID1	Q1	X1	Y1	Z1	Vx1	Vy1	Vz1
10:04:35.25	2	1	64	0.00	-0.02	0.67	1.76	-0.29	2.33	0.15
10:04:35.65	2	1	64	0.00	-0.07	0.95	1.74	-0.32	2.35	-0.17
10:04:36.05	2	1	64	0.00	-0.12	1.18	1.72	-0.36	2.33	-0.30
10:04:36.45	2	1	64	0.00	-0.16	1.36	1.69	-0.38	2.21	-0.41

Figura 7 – File CSV come risultato dello script, con coordinate e velocità del soggetto.

Lo scopo principale di questo script è automatizzare il processo di estrazione, conversione e pulizia dei dati esadecimali grezzi, rendendoli facilmente analizzabili in un formato CSV strutturato in questo caso per le coordinate relative alla persona (TLV=2), lo script potrà essere modificato in modo agile a seconda delle richieste.

## Elaborazione dei dati – NUVOLA DI PUNTI (DEBUG = 2)

Con la modalità DEBUG=2 il radar genera dei dati leggermente diversi da quelli visti per la modalità debug 3. In questo caso viene tenuto conto anche della nuvola di punti che circonda/rappresenta ogni individuo all' interno della stanza. Infatti come si può notare nel manuale :

Field		Number of bytes	Description
HEAD		8	Frame header, fixed\x01\x02\x03\x04\x05\x06\x07\x08
LENGTH		4	Whole frame data length (uint32)
FRAME		4	Frame number (uint32)
TLVs		4	TLVs=1 followed by point cloud information (uint32)
POINTLENTH		4	Point cloud data length (points = POINTLENTH/25) (uint32)
Point 1	x	4	Coordinates x/y/z and energy signal-to-noise ratio information (v is int8, all others are float)
	y	4	
	z	4	
	v	1	
	SNR	4	
	PO W	4	
	DPK	4	
.....			
Point n	x	4	Coordinates x/y/z and energy signal-to-noise ratio information (v is int8, all others are float)
	y	4	
	z	4	
	v	1	
	SNR	4	
	PO W	4	
	DPK	4	
TLVs		4	TLVs=2 followed by person information (uint32)
TRACKLENTH		4	Length of person data (number of persons = TRACKLENTH/32) (uint32)
Personnel 1	ID	4	Personnel markers (uint32)
	Q	4	(uint32)
	X	4	X/Y/Z coordinates of the person and the speed (float)
	Z	4	
	Y	4	
	Vx	4	
	Vz	4	
	Vy	4	
.....			
Personnel n	ID	4	Personnel markers (uint32)
	Q	4	(uint32)
	X	4	X/Y/Z coordinates of the person and the velocity (float), in units: coordinates in m and velocity in m/s, to two decimal places. Single precision floating point type according to the standard for binary floating point arithmetic (IEEE 754),with the small endunwrappedbefore. <a href="https://www.binaryconvert.com/convert_float.html">https://www.binaryconvert.com/convert_float.html</a> <a href="https://www.cnblogs.com/guanshan/articles/guan022.html">https://www.cnblogs.com/guanshan/articles/guan022.html</a>
	Z	4	
	Y	4	
	Vx	4	
	Vz	4	
	Vy	4	

Figura 8 – Parametri e descrizione DEBUG 2



Ovviamente il file di testo generato da SSCOM è diverso da quello visto in precedenza, la dimensione di ogni riga del file è molto più variabile rispetto a prima dato che i punti che compongono ogni persona ad ogni istante possono essere più di 30.

```
[11:28:04.637]IN;û;ô01 02 03 04 05 06 07 08 A6 05 00 00 AC 00 00 00 01 00 00 00 46 05 00
00 01 5E F5 BE 01 24 01 BF EF 71 BA 3F FE EC 88 B6 41 00 00 80 3F 10 45 0A 41 01 5E F5 BE
01 99 07 BF A2 4B B9 3F FF 5E 76 39 42 00 00 80 3F C6 80 0B 41 01 24 01 BF 01 0E 0E BF 7C
F9 B6 3F 00 64 EF 2F 42 00 00 80 3F 22 6D 10 41 01 99 07 BF 01 5E F5 BE A2 4B B9 3F 01 00
00 80 3F DE 92 E7 40 01 DA 17 BF 01 E0 DC BE 09 EE C7 3F 01 8A A7 0D 42 00 00 80 3F 15
07 03 41 01 00 E4 BE 01 40 39 BF 80 63 C8 3F FD 92 AE C3 40 00 00 80 3F 6C 68 C0 40 01
00 E4 BE 01 A0 15 BF 77 B2 CF 3F FE 78 35 E4 41 00 00 80 3F 45 09 E9 40 01 60 07 BF 01 ...
... .. 00 00 00 00
```

Lo script python precedente ovviamente non va più bene per questo nuovo caso, ne ho realizzato un altro chiamato "*pointCloud.py*" che gestisce le coordinate relativi alla nuvola di punti. Ci sono molti passaggi intermedi per l'elaborazione del file, ma sostanzialmente quello che otteniamo alla fine è un file CSV dove per ogni istante di secondo vengono calcolate le media fra tutti i punti relativi alle X,Y,Z della nuvola. Avendo in questo modo un riferimento ulteriore alla presenza/posizione della persona.

## Utilizzo dello script

Per eseguire lo script, digitare dalla riga di comando:

```
> python pointCloud.py <path/to/input_file.TXT> <risultatoFinale.csv>
```

Il flow che c'è dietro a tutta l'esecuzione dello script possiamo vederlo in questa immagine :

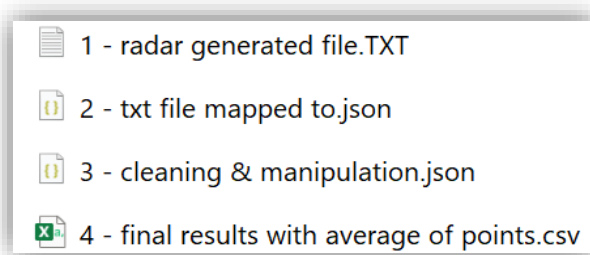


Figura 9 – Esecuzione *pointCloud.py* script

In questo caso ho dovuto mappare in un file json tutti i punti per ogni istante, dato che variano continuamente non potevo più usare il csv, le conversioni dei byte rimangono invariate come nello script precedente. Vengono generati più file e non solo quello con i risultati finali in csv, potrebbero tornare utili per lavorarci sopra.

```
{
  "time": "11:27:13.023",
  "headers": [
    "IN\u00a1\u00fb\u00a1\u00f401",
    "02",
    "03",
    "04",
    "05",
    "06",
    "07",
    "08"
  ],
  "frame": [
    "84",
    "00",
    "00",
    "00"
  ],
  "TLV1": "01000000",
  "PointL": "64000000",
  "points": [
    {
      "x1": "017C3FBE",
      "y1": "0135DCBF",
      "z1": "F4A9D33F",
      "v1": "04",
      "SNR1": "30346140",
      "POW1": "ED1DFBEC",
      "DPK1": "7FF6E040"
    },
    {
      "x2": "02D0F5BD",
```

Figura 10 – file .txt mappato in JSON

```
{
  "time": "11:27:13.023",
  "points": [
    {
      "x1": "-0.19",
      "y1": "-1.72",
      "z1": "1.65"
    },
    {
      "x2": "-0.12",
      "y2": "-1.80",
      "z2": "1.82"
    },
    {
      "x3": "-0.16",
      "y3": "-1.59",
      "z3": "2.07"
    },
    {
      "x4": "-0.11",
      "y4": "-2.76",
      "z4": "2.43"
    }
  ]
},
```

Figura 11 – manipolazione file JSON (conversione bytes)

	A	B	C	D
1	time	average_x	average_y	average_z
2	11:27:13.0	-0.14	-1.97	1.99
3	11:27:13.8	-0.17	-1.73	1.92
4	11:27:14.2	-0.12	-1.79	1.94
5	11:27:14.6	0.24	-1.57	1.86
6	11:27:15.0	-0.15	-1.65	1.73
7	11:27:15.4	-0.11	-1.81	1.88
8	11:27:15.8	-0.15	-1.77	2.05

Figura 12 – csv finale con medie dei punti

## Visualizzazione punti 2D e 3D

Il software *Radar\_DemoSideMount.exe* ci consente di visualizzare in 2D e 3D quello che succede all'interno del raggio d'azione del radar.

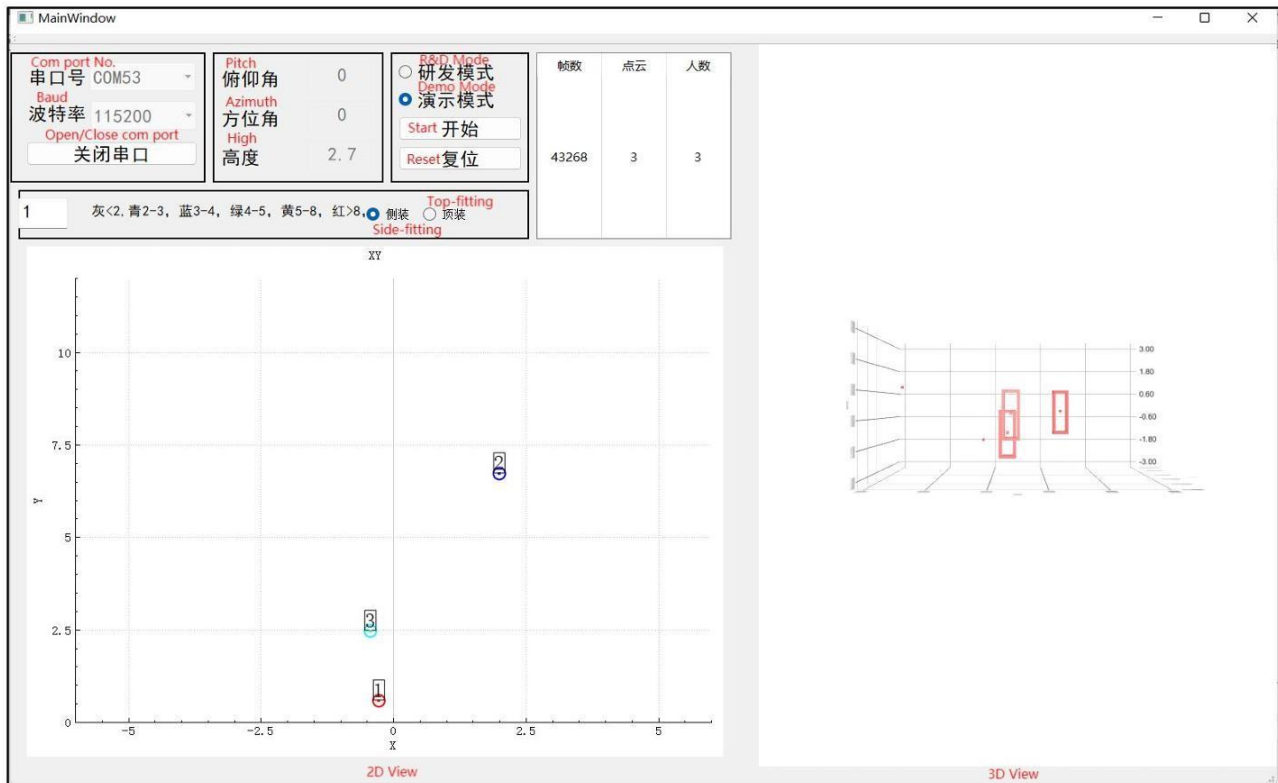


Figura 13

I passaggi per farlo funzionare correttamente sono riportati di seguito :

- 1) Connettere USBtoTTL il radar, poi aprire il software "Radar\_DemoSideMount.exe" (verificare che sia stato impostato in modalità dati HEX (AT+DEBUG=2/3));
- 2) Selezionare il numero di porta seriale come mostrato in Figura 3, il baud rate predefinito è 115200, quindi cliccare su "Apri Porta Seriale";
- 3) Selezionare "Top or Side Fit";
- 4) Cliccare su "Start" e il radar inizierà a funzionare;
- 5) Selezionando "R&D Mode" verrà visualizzata la nuvola di punti, mentre selezionando "Demo Mode" non verranno visualizzati i dati della nuvola di punti;
- 6) Come mostrato in Figura 3, il lato sinistro visualizza le coordinate 2D, mentre il lato destro visualizza la visualizzazione 3D;
- 7) Cliccare sul pulsante "Stop" per interrompere il funzionamento del radar.