77 GHz Millimeter Wave Radar Sensor Protocols

Model No: CAR-A60

1. Interface Overview

A60 77G radar sensor supports CAN interface, CAN bus communication network conforms to ISO11898-2 standard, transmission rate is 500K bit/s. The A60 sensor transmits 77G signals to the perimeter, receives the return signals, and then processing received data,calculate to obtain the trajectory information of the target group.

The CAN interface of A60 is used to configure sensors, or input vehicle speed and angular speed information, output sensor status and target information. A CAN bus can mount up to 8pcs A60 sensors. You can configure the sensor ID and output data corresponding to MessageIDs.

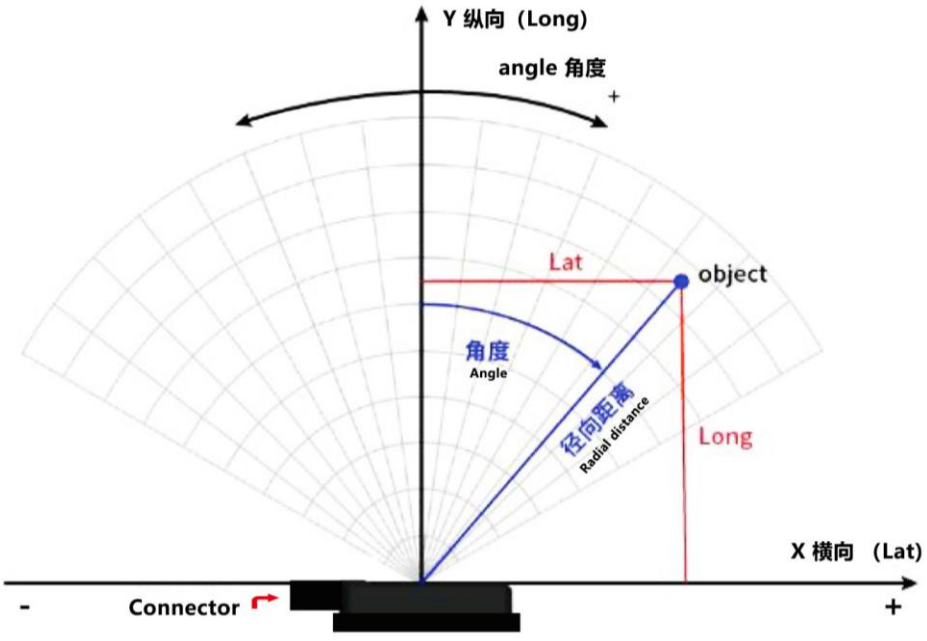
The sensor ID ranges from 0 to 7. The message ID is calculated as follows:
MsgID=MsgID_0+SensorID*0x10
For example,
Sensor ID=0, the configuration message ID is 0x200.
If the sensor ID is 1, the configuration message is 0x210, and so on.
After the sensor ID is set, the sensor will only respond to the new configuration message.

Sensor CAN message (SensorID=0)

In/Out	ID	Message Name	Content	Section
In	0x200	RadarCfg	Radar configuration info	
In	0x401	Region configuration	Region configuration	
Out	0x201	RadarState	Radar status info	
In	0x300	SpeedInformation	Speed info(unused)	
In	0x301	Yaw rate Information	Yaw rate Info(unused)	
Out	0x60A	Obj_0_Status	Target status info	
Out	0x60B	Obj_1_General	Target general info	
Out	0x700	SoftWareVersion	Software version	
Out	0x402	Region state	Region setting status	

2. Radar Introduce

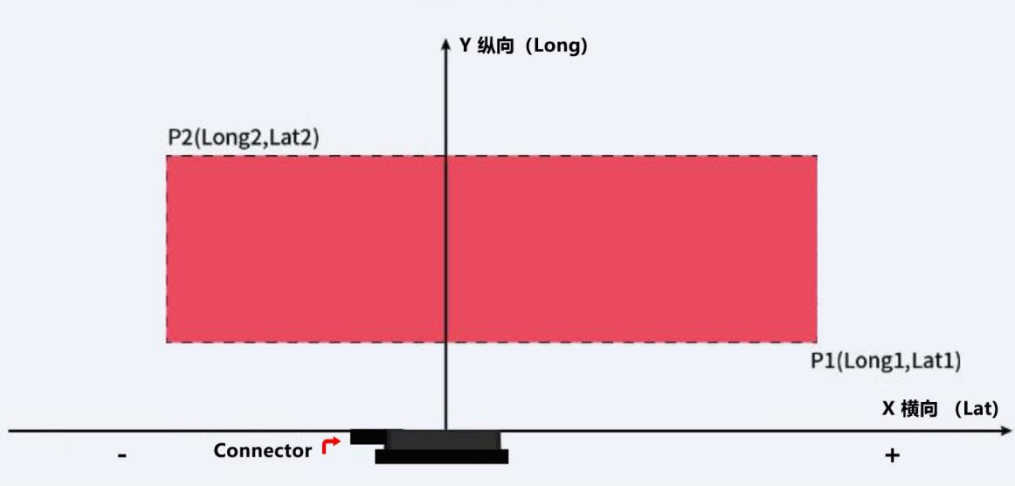
The A60 radar sensor uses 77GHz high-frequency Millimeter waves to analyze its surroundings. After the reflected signal is processed in several steps, the data is output as Objects. Objects contain historical track and object detection information. Where, the velocity of the target approaching the radar is negative (-), while the velocity of the target away from the radar is positive (+).
Radar Cartesian coordinates are shown below:



✧ ***By default, the output of target data is based on the radial distance of the target from near to far.***

3. Detect Area Settings

A60 radar sensor CAN protocol Area/region detect setting function, default output rectangle box (80*50 meters), the target according to the radial distance from near to far order output.
The rectangle box is set as follows:



Currently, only a single input box can be set, that is, two coordinate points P1 and P2 determine a rectangular box, and the number of output targets in the rectangular box can be set. Suppose that the number of targets allowed to be output in the rectangle box is set as N. If N is larger than the number of targets actually detected in the rectangle box, the number of targets actually detected will be output; if N is smaller than the number of targets actually detected, N nearest targets will be output (sorted by radial distance, from near to far).

4. Configure the input

4.1 Configuration Message (0x200)

Configure basic radar parameters using Message RadarCfg 0x200. You do not need to periodically set configuration parameters. If RadarCfg_StoreInNVM is set to 1, store the parameters to non-volatile storage (NVM) and then enable the new configuration parameters automatically at power-on startup. If the parameters are not saved to the NVM, the previous configuration parameters are invalid during the next power-on. Minimize the number of non-volatile memory Settings. Frequent Settings shorten the memory service life.

	7	6	5	4	3	2	1	0
0	Nvm valid 7	Index valid 6	Extinfo valid 5	Quality valid 4	Type valid 3	Power valid 2	SensorID valid 1	MaxDistance valid 0
1	MaxDistance 5 msb	14	13	12	11	10	9	8
2	MaxDistance 3	22 lsb	21	20	19	18	17	16
3	31	30	29	28	27	26	25	24
4	RadarPower 39 msb	38	37 lsb	OutputType 36 msb	35 lsb	SensorID 34 msb	33	32 lsb
5	StoreNVM 47	SortIndex 46 msb	45	44 lsb	SendExtInfo 43	SendQuality 42	41	40
6	55	Interface Select Valid 54	Interface type 53 msb	52 lsb	RCS Threshold 51 msb	50	49 lsb	RCS Threshold Valid 48
7	RadarCfg BaudRate 63 msb	62	61 lsb	RadarCfg BaudRate valid 60	59	58	57	56

Signal	Start	Len	Min	Max	Res	Uint/Define
RadarCfg_MaxDistance_Valid	0	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_SensorID_Valid	1	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_RadarPower_Valid	2	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_OutputType_Valid	3	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_SendQuality_Valid	4	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_SendExtInfo_Valid	5	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_SortIndex_Valid	6	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_StoreInNvm_Valid	7	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_MaxDistance	22	10	0	2048	2	m
RadarCfg_SensorID	32	3	0	7	1	ID(0 to 7) (Default ID is 0)
RadarCfg_OutputType	35	2	0	2	1	0x0:none, 0x1:Objects (default), 0x2:Clusters
RadarCfg_RadarPower	37	3	0	7	1	0x0:Standard (default)
RadarCfg_SendQuality	42	1	0	1	1	0x0:Inactive,0x1:Active
RadarCfg_SendExtInfo	43	1	0	1	1	0x0:Inactive,0x1:Active
RadarCfg_SortIndex	44	3	0	7	1	0x0:No Sorting, 0x1:Sorted by rang (default), 0x2:Sorted by RCS
RadarCfg_StoreNVM	47	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_RCS_Threshold_Valid	48	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_RCS_Threshold	49	3	0	7	1	0x0:Standard (default), 0x1:HighSensitivity
RadarCfg_BaudRate_Valid	60	1	0	1	1	0x0:Invalid, 0x1:Valid
RadarCfg_BaudRate	61	3	0	7	1	0x0:500K(default), 0x1:250K, 0x2:1M

Remarks:

1. The Objects mode is sorted by distance by default.

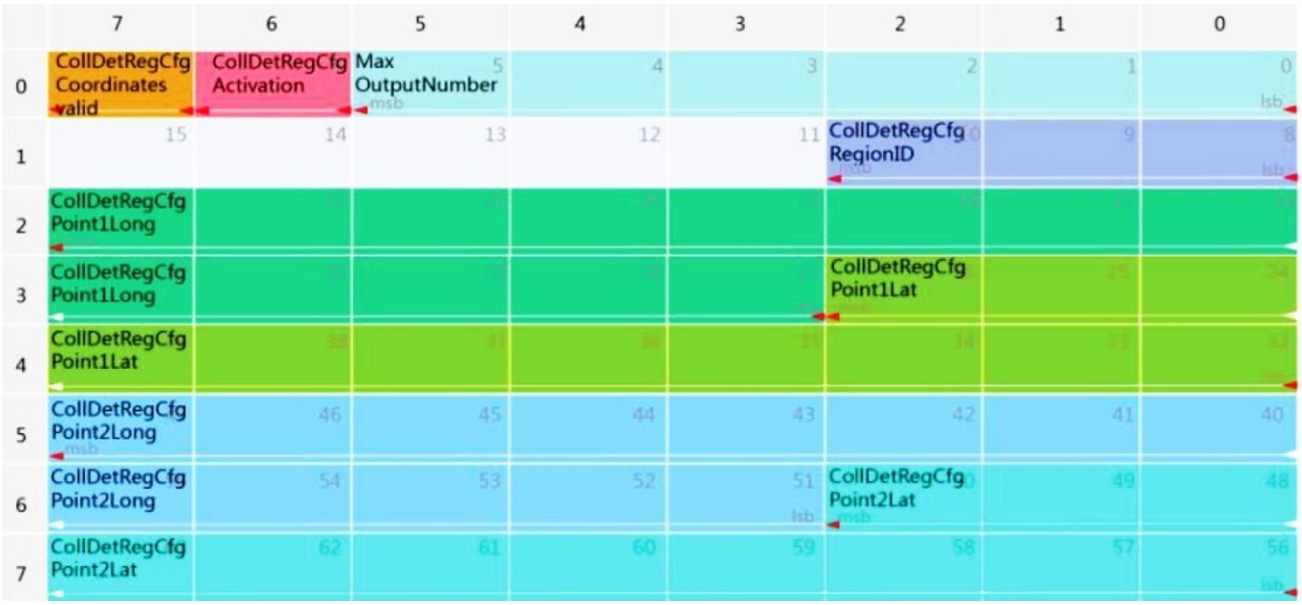
2. Enable RadarCfg_StoreNVM. The saved parameters take effect after the power failure and restart.

3. res indicates resolution. For example, if the maximum distance value is set to 25, the resolution is 2, then the maximum distance value is 50 meters.

4. The maximum distance setting is not open.

4.2 Set the number of Rectangle Boxes and target 0x401

collision detection region configuration(0x401) Currently, only one detection region can be set. The command format is as follows:



Signal	Start	Len	Offset	Min	Max	Res	Descrition
Max_OutputNumber	0	6		0	63	1	Maximum number of output targets
CollDetRegCfg_Activation	6	1		0	1	1	0x0:Inactive,0x1:Active
CollDetRegCfg_ CoordinatesValid	7	1		0	1	1	0x0:Invalid, 0x1:Valid
CollDetRegCfg_RegionID	8	3		0	7	1	Area ID. The default value is 1
CollDetRegCfg_Point1Long	27	13	-500	-500	1138.2	0.2	Unit: m
CollDetRegCfg_Point1Lat	32	11	-204.6	-204.6	204.8	0.2	Unit: m
CollDetRegCfg_Point2Long	51	13	-500	-500	1138.2	0.2	Unit: m
CollDetRegCfg_Point2Lat	56	11	-204.6	-204.6	204.8	0.2	Unit: m

- ✧ Max_OutputNumber: the maximum number of objects that can be outputed in the set rectangle is N≦64. After N was set(output number you want), and If n is stand for objects, when n(object numbers) larger than N, then output N objects' data, outputed from near to far according to the radial distance. If the actual number of targets detected n<N, then output the actual number of object's data.

- ✧ CollDetRegCfg_Activation: Indicates whether collision detection is activated. 0x0:inactive, 0x1:active. After this function is enabled and the coordinates are valid, the subsequent rectangle Settings take effect. That is, re-power-on, the parameters set last time are valid.

- ✧ CollDetRegCfg_CoordinatesValid: The coordinate point is set to 0x0:invalid and 0x1:active.
  The collision detection function is enabled and also the coordinate is valid, then coordinate set can be success. Otherwise, the coordinate setting does not take effect.
  Coordinates are valid but inactive: 0x04 have no 0x402 Output.
  Coordinates are invalid , active: 0x02,0x402 output coordinates are all 0x00, region ID are valid.

- ✧ CollDetRegCfg_Point1Long: The longitudinal (Y)distance of coordinate point 1;
- ✧ CollDetRegCfg_Point1Lat: Transverse (X)distance value at coordinate point 1;
- ✧ CollDetRegCfg_Point2Long: The longitudinal (Y)distance of coordinate point 2;
- ✧ CollDetRegCfg_Point2Lat: Transverse(X) distance value of coordinate point 2;

Note for setting coordinate points: Coordinate point 1 is the coordinate value of the lower right corner of the rectangle box, and coordinate point 2 is the coordinate value of the upper left corner of the rectangle box.

To be satisfied,

CollDetRegCfg_Point1Long<CollDetRegCfg_Point2Long

CollDetRegCfg_Point1Lat>CollDetRegCfg_Point2Lat

Otherwise, the rectangle box is invalid and will not be saved to flash.To be satisfied,

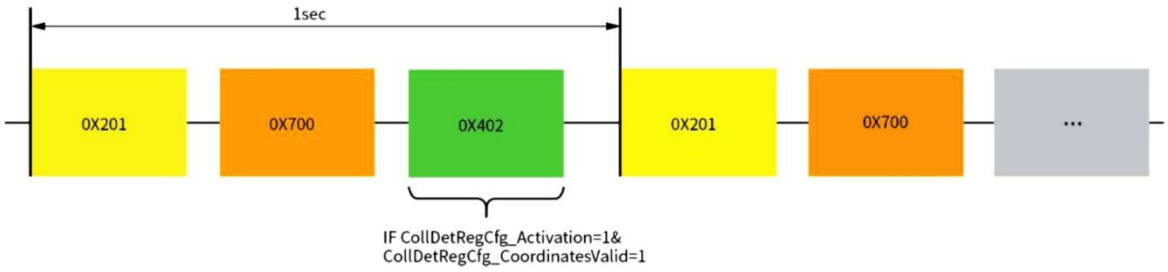
CollDetRegCfg_Point1Long<CollDetRegCfg_Point2Long

CollDetRegCfg_Point1Lat>CollDetRegCfg_Point2Lat

Otherwise, the rectangle box is invalid and will not be saved to flash.

5. Radar Status output

The radar periodically (1 second) sends the radar configuration and status information through Message 0x201 and the radar firmware information through 0x700. If detect area is enabled and the coordinate setting is valid, the output is 0x402. If either area setting or coordinate setting is invalid, the radar outputs objects mode object data.



Periodically sends status information

5.1 Radar Status Information (0x201)

The radar periodically (every second) sends status information. After the configuration of radar parameters takes effect, the radar status message immediately sends the command 0x201 to prove that the configuration parameters have taken effect.

	7	6	5	4	3	2	1	0
0	NVM-WriteStatus 7	NVM-ReadStatus 6	5	4	3	2	1	0
1	Max-DistanceCfg 15 msb	14	13	12	11	10	9	8
2	Max-DistanceCfg 23	22 lsb	21	20	19	18	17	16
3	31	30	29	28	27	26	RadarPower Cfg 25	24
4	RadarPower Cfg 39	SortIndex 38 msb	37	36 lsb	35	SensorID 34 msb	33	32 lsb
5	Motion-RxState 47 msb	46 lsb	SendExtInfo-Cfg 45	SendQuality-Cfg 44	OutputType-Cfg 43 msb	42 lsb	41	40
6	Can BaudRate 55 msb	54	53 lsb	52	51	50	49	48
7	63	62	61	RCS threshold 60 msb	59	58 lsb	Interface type 57 msb	56 lsb

Signal	Start	Len	Min	Max	Res	Uint/Define
RadarState_NVMReadStatus	6	1	0	1	1	0x0:failed,0x1:Successful
RadarState_NVMWriteStatus	7	1	0	1	1	0x0:failed,0x1:Successful
RadarState_MaxDistanceCfg	22	10	0	2046	2	m
RadarState_SensorID	32	3	0	7	1	Current radar ID(0~7)
RadarState_SortIndex	36	3	0	7	1	0x0:no sorting,0x1:sorted by range,0x2:sorted by RCS
RadarState_RadarPowerCfg	39	3	0	7	1	0x0:Standard
RadarState_OutputTypeCfg	42	2	0	3	1	0x0:none,0x1:Objects, 0x2:Clusters
RadarState_SendQualityCfg	44	1	0	1	1	0x0:Inactive,0x1:Active
RadarState_SendExtInfoCfg	45	1	0	1	1	0x0:Inactive,0x1:Active
RadarState_MotionRxState	46	2	0	1	1	0x0:input ok; 0x1:speed missing; 0x2:yaw rate missing; 0x3:speed and yaw rate issing
RadarState_CANBaudRate	53	3	0	1	1	0x0:500K(default),0x1:250K, 0x2:1M
RadarState_RCS_threshold	58	3	0	1	1	0x0:Standard; 0x1:High sensitivity

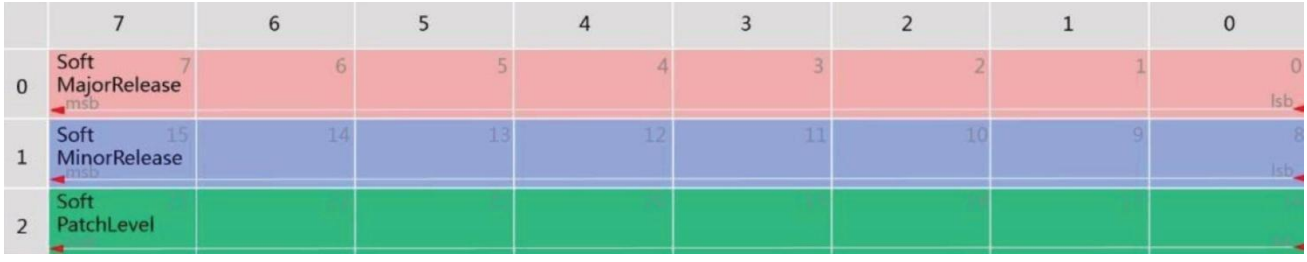
5.2 Collision detection area state (0x402)

In 0x401,when CollDetRegCfg_Activation=1 and also CollDetRegCfg_CoordinatesValid=1 , the radar periodically sends a collision detection status message (1s).

When neither CollDetRegCfg_Activation nor CollDetRegCfg_CoordinatesValid value is 1,radar outputs objects mode targets.

Signal	Start	Len	Offset	Min	Max	Res	Descrition
Max_OutputNumber	0	6		0	63	1	Maximum number of output targets
CollDetRegState_RegionID	8	3		0	7	1	The default area number is 1
CollDetRegState_Point1Long	19	13	-500	-500	1138.2	0.2	Unit: m
CollDetRegState_Point1Lat	34	11	-204.6	-204.6	204.8	0.2	Unit: m
CollDetRegState_Point2Long	43	13	-500	-500	1138.2	0.2	Unit: m
CollDetRegState_Point2Lat	48	11	-204.6	-204.6	204.8	0.2	Unit: m

5.3 Heartbeat message (0x700)



Signal	Start	Len	Min	Max	Res	Uint/Define
Soft_MajorRelease	0	8	0	255	1	Software main verion
Soft_MinorRelease	8	8	0	255	1	Software mirror verion
Soft_PatchLevel	16	8	0	255	1	Software debug verion

6. Object list mode output

The Object list contains two messages that are sent periodically.

1. Object_0_Status(0x60A) First list header message, which contains information about the number of targets to be sent.

2. Object_1_General(0x60B) This message contains the speed and distance information of the target. Periodically send all tracked target information.

6.1 Object List Status(0x60A)

The Object List Status (0x60A) contains the target list header message, which is first sent at 0x60A during each measurement period.

Signal	Start	Len	Min	Max	Res	Uint/Define
Objects_NofObjects	0	8	0	255	1	Number of objects
Objects_MeasCount	8	16	0	65535	1	Cycle count, cycle plus 1
Objects_InterfaceVersion	28	4	0	15	1	Target list CAN interface version

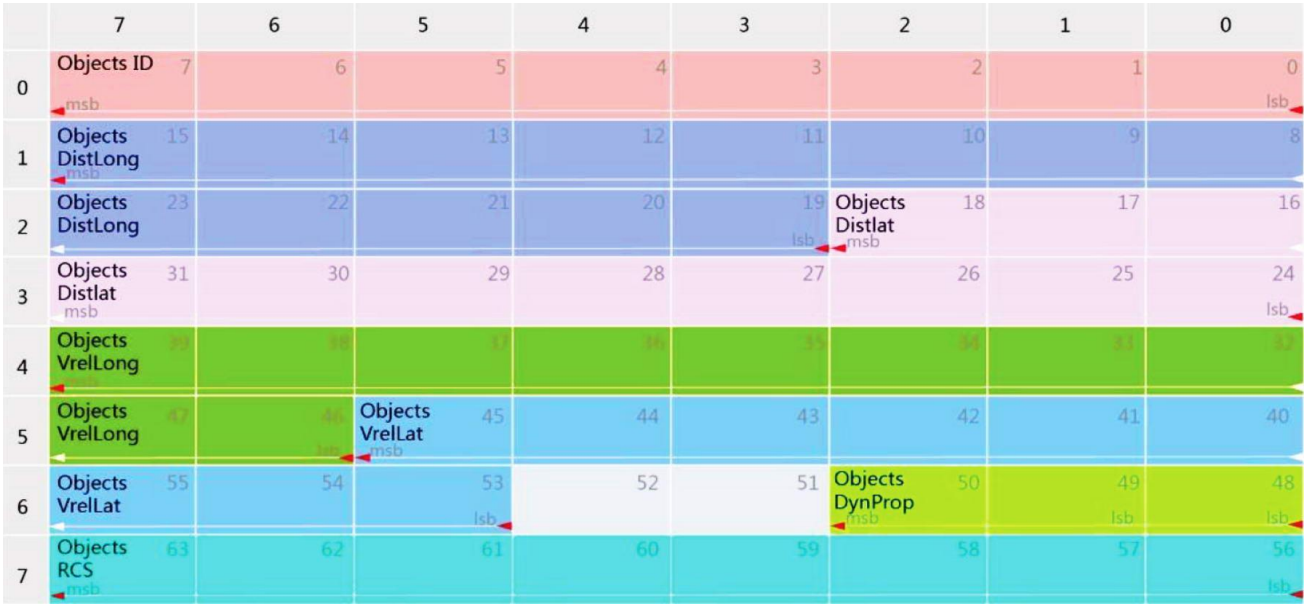
Objects_NofObjects: Number of objects;

Objects_MeasCount: Cycle count from 0 to 65535;

Objects_InterfaceVersion: CAN interface version. The default value is 0.

6.2 Objects General Information (0x60B)

This message contains information about the location and speed of the target and periodically sends all tracking target information.



Signal	Start	Len	Offset	Min	Max	Res	Descrition
Objects_ID	0	8		0	255	1	
Objects_DistLong	19	13	-500	-500	1138.2	0.2	m
Objects_Distlat	24	11	-204.6	-204.6	204.8	0.2	m
Objects_VrelLong	46	10	-128	-128	127.75	0.25	m/s
Objects_DynProp	48	3		0	7	1	0x0:moving; 0x1:stationary; 0x2:oncoming; 0x3:stationary candidate; 0x4:unknown; 0x5:crossing stationary; 0x6:crossing moving; 0x7:stopped
Objects_VrelLat	53	9	-64	-64	63.75	0.25	m/s
Objects_RCS	56	8	-64	-64	63.5	0.5	dBm^2(reserve)

Objects_ID: object ID;
Objects_DistLong: object longitudinal distance;
Objects_Distlat: object horizontal distance;
Objects_VrelLong: target longitudinal velocity;
Objects_DynProp: Dynamic attribute of the target. The default value is 0. Classification is not supported at the moment.
Objects_VrelLat: target lateral velocity;
Objects_RCS: target RCS.

Target radial distance:

R=sqrt(Objects_DistLong*Objects_DistLong+Objects_Distlat*Objects_Distlat)

The target Angle is:

$$\text{Tan}\theta = \frac{\text{Objects\_Distlat}}{\text{Objects\_DisLong}}$$

The target velocity is *V*,

$$V = \text{Objects\_VrelLong} * \cos\theta + \text{Objects\_VrelLat} * \sin\theta$$

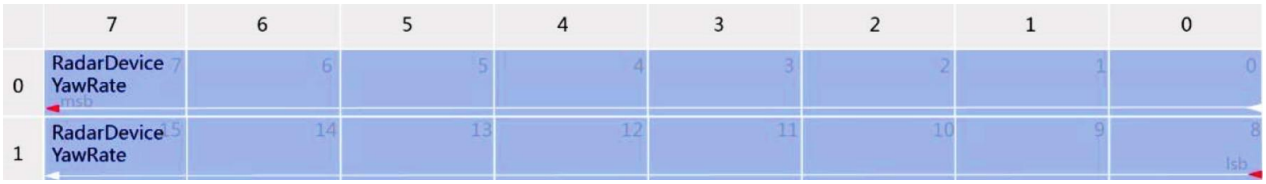
7. Sports information input (reserved)

The sensor receives input messages 0x300 and 0x301, but still works without input.
After 500ms, the sensor status is as follows:
1. speed0m/s is stationary;
2, angular velocity 0deg/s;
Each message has independent timeout monitoring. The timeout state is reflected in the RadarState_MotionRxState bit of the 0x201 message.

7.1 Speed Information(0x300)

Signal	Start	Len	Min	Max	Res	Uint/Define
Radardevice_speedDirection	6	2	0	2	1	0x0: Static 0x1: Forward 0x2: Backward
RadarDevice_Speed	8	13	0	163.8	0.02	m/s

7.2 Yaw Rate Information(0x301)



Signal	Start	Len	Min	Max	Res	Uint/Define
Radardevice_speedDirection	8	16	-327.7	327.68	0.01	deg/s

8. Protocol parsing example

Res: resolution.
The received radar data multiplied by this value is the actual target value.
For example, if the received value is 100 and the resolution is 0.2, then actual value is 100*0.2=20.

Min: offset.
Some values may have negative values (-). In order to ensure that the output is positive (+) and easy to parse, the radar add the offset. When analyzing radar data, you need to add offset.
If the radar sent data is Value1, then actual value is Value2=Value1*Res+Min.

8.1 Message Configuration Example (0x200)

The program defaults radar_outputType as Objects, modifies the storage parameters to NVM, radar_power as standard, SortIndex as range, RCS_Threshold as standard, SensorID as 0, The MaxDistance is 160 meters.

For example, if the message ID is 0x200, the message content is 82 00 00 00 01 80 00 00
This message modifies the radar ID to 1 and saves the modified parameter to NVM, that is, when the radar is powered on next time, the ID is 1.
The configuration takes effect only when the Valid bit is enabled. Otherwise, the configuration does not take effect.

8.2 Heartbeat Signal (0x700)

The first three bytes of the message represent the software version number. If the message returned is 0x700, the current radar ID is 0, and 0x710, the current radar ID is 1.

For example, if the return message is 0x01 0x00 0x15 0x00 0x00 0x00 0x00 The current software version of the radar is V1.0.21.

8.3 Configuration command

After the radar ID is configured, the output MessageIDs are also changed. The sensor ID ranges from 0 to 7. The message ID can be calculated as follows:

MsgID=MsgID_0+SensorID*0x10

For example:
1. Change the radar id to 1 and save it. The command is: 8200000001800000
2. Change the radar id to 2 and save it. The command is: 8200000002800000
3. Change the radar id to 3 and save it. The command is: 8200000003800000

8.4 Target information (0x60B)

The message message is: **0x57 0x4E 0xC4 0x0C 0x7F 0x60 0x00 0x80**
That is:

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
0x57	0x4E	0xC4	0x0C	0x7F	0x60	0x00	0x80

1. Target ID: 0x57=87, that is, the target ID is 87. The target is generated iteratively between 0 and 255. In the stable tracking process, the target ID remains unchanged;

2. Longitudinal distance （Y）of target: (0x4E*32+0xC4>>3)*0.2-500=4 m;
3. Transverse distance （X）of target: 【(0xC4&0x07)*256+0x0C】*0.2-204.6= 2.6m;
4. Target longitudinal （Y）velocity: (0x7F*4+0x60>>6)*0.25-128=-0.75m/s;
5. Target lateral （X）velocity: 【(0x60&0x3F)*8+0x00>>5】*0.25-64=0m/s;
6. Target dynamic attribute: 0x00&0x07=0, the default is 0;
7. RCS: 0x80*0.5-64=0, all default values are 0.

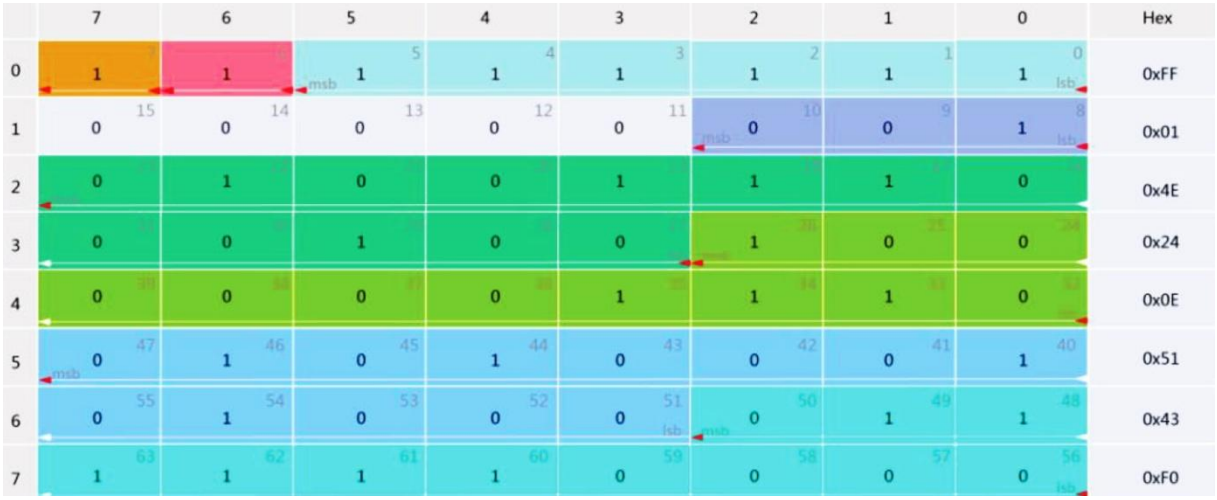
8.5 Region Settings (0x401)

（8.5.1）
If you need to set the rectangle box to 6*20 meters and the maximum number of output objects to 63, you need to enable the region setting and coordinates.

Suppose the coordinate of point 1 is: (0,3), the coordinate of point 2 is: (20, -3),
Then sending command is: **0xFF 0x01 0x4E 0x24 0x0E 0x51 0x43 0xF0**

CollDetRegState_Point1Long=(0+500)x5=2500=(100111000100)b
CollDetRegState_Point1Lat=(3+204.6)x5=1038=(10000001110)b
CollDetRegState_Point2Long=(20+500)x5=2600=(101000101000)b
CollDetRegState_Point2Lat=(-3+204.6)x5=1008=(1111110000)b

The analysis is as follows:



(8.5.2)

If you need to set the rectangle box to 10*50 meters and the maximum number of output objects to 63, you need to enable the region setting and coordinates.

Suppose the coordinates of point 1 are: (0,5), the coordinates of point 2 are: (50, -5),
Then sending command is: **0xFF 0x01 0x4E 0x24 0x18 0x55 0xF3 0xE6**

CollDetRegState_Point1Long=(0+500)x5=2500=(100111000100)b
CollDetRegState_Point1Lat=(5+204.6)x5=1048=(10000011000)b
CollDetRegState_Point2Long=(50+500)x5=2750=(101010111110)b
CollDetRegState_Point2Lat=(-5+204.6)x5=998=(1111100110)b

The analysis is as follows:

	7	6	5	4	3	2	1	0	Hex
0	1	1	1	1	1	1	1	1	0xFF
1	0	0	0	0	0	0	0	1	0x01
2	0	1	0	0	1	1	1	0	0x4E
3	0	0	1	0	0	1	0	0	0x24
4	0	0	0	1	1	0	0	0	0x18
5	0	1	0	1	0	1	0	1	0x55
6	1	1	1	1	0	0	1	1	0xF3
7	1	1	1	0	0	1	1	0	0xE6

Region/Area setting Note:

1) Only when coordinate points 1 and 2 meet the condition:
[(point1Long<point2Long)&&(point1Lat>point2Lat)],
then parameter will be saved; otherwise, the parameter will not be saved;

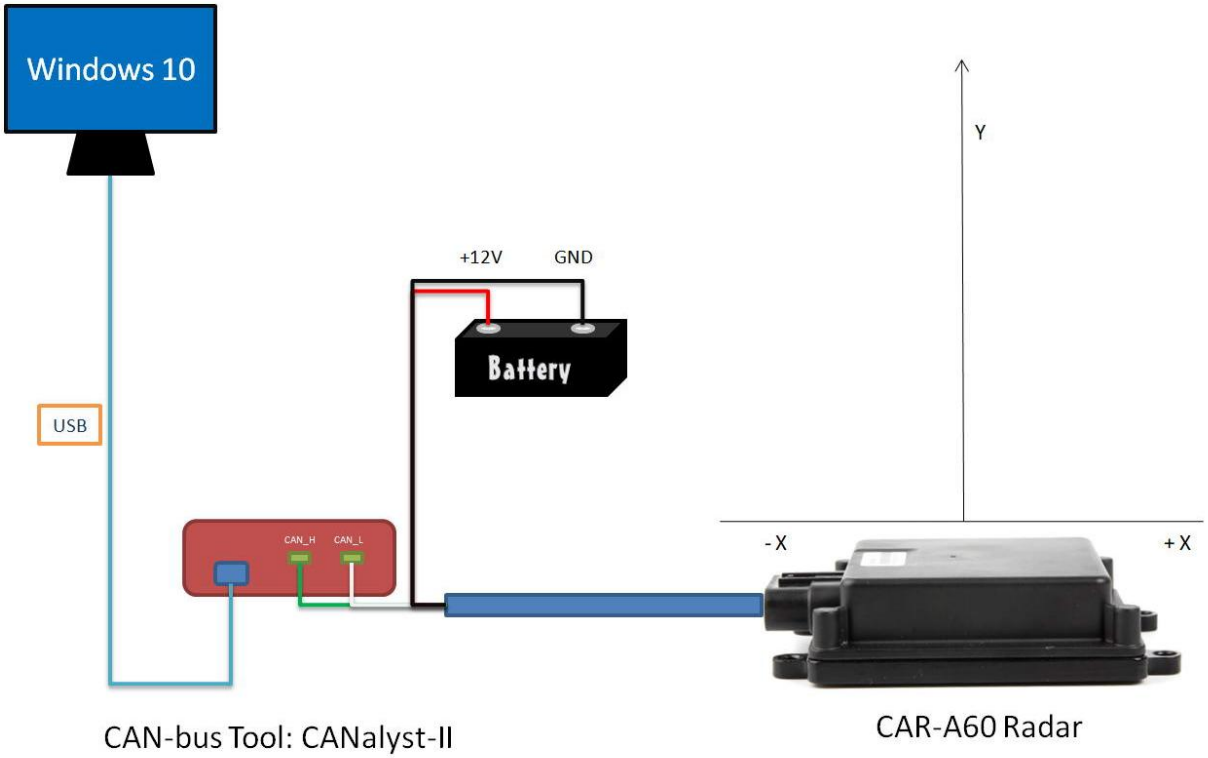
2) As long as the coordinates meet the conditions, save the parameters to Flash regardless of whether the region setting is activated and the coordinates are valid;

3) But only when the locale setting and coordinates are valid, the rectangle box will be drawn, 0x402 periodic output;

4) Region Settings and coordinates As long as one of them is invalid, the radar enters the objects multi-object mode to output all targets, i.e. rimless, no 0x402 output.

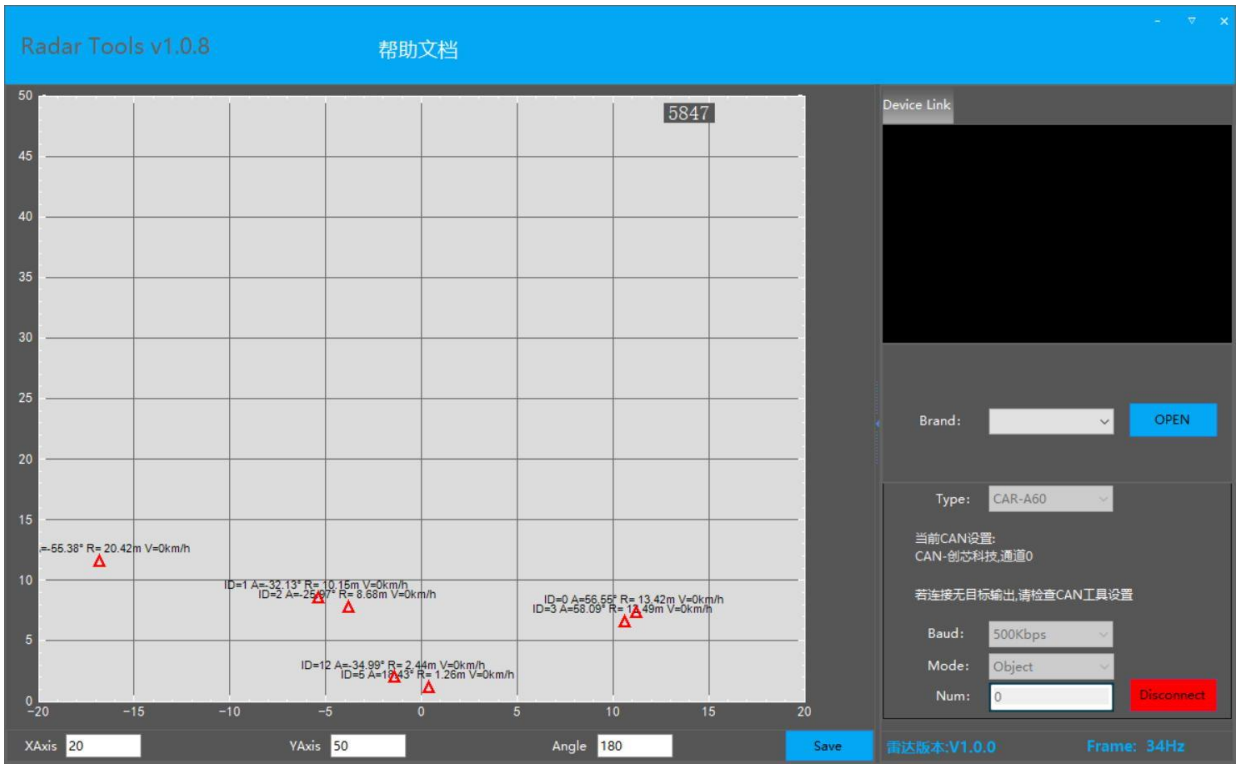
9. Hardware connection and test.

- CAN Interface, BaudRate 500K
- Red: +8~26VDC
- Black: GND
- Green: CAN_H
- White: CAN_L

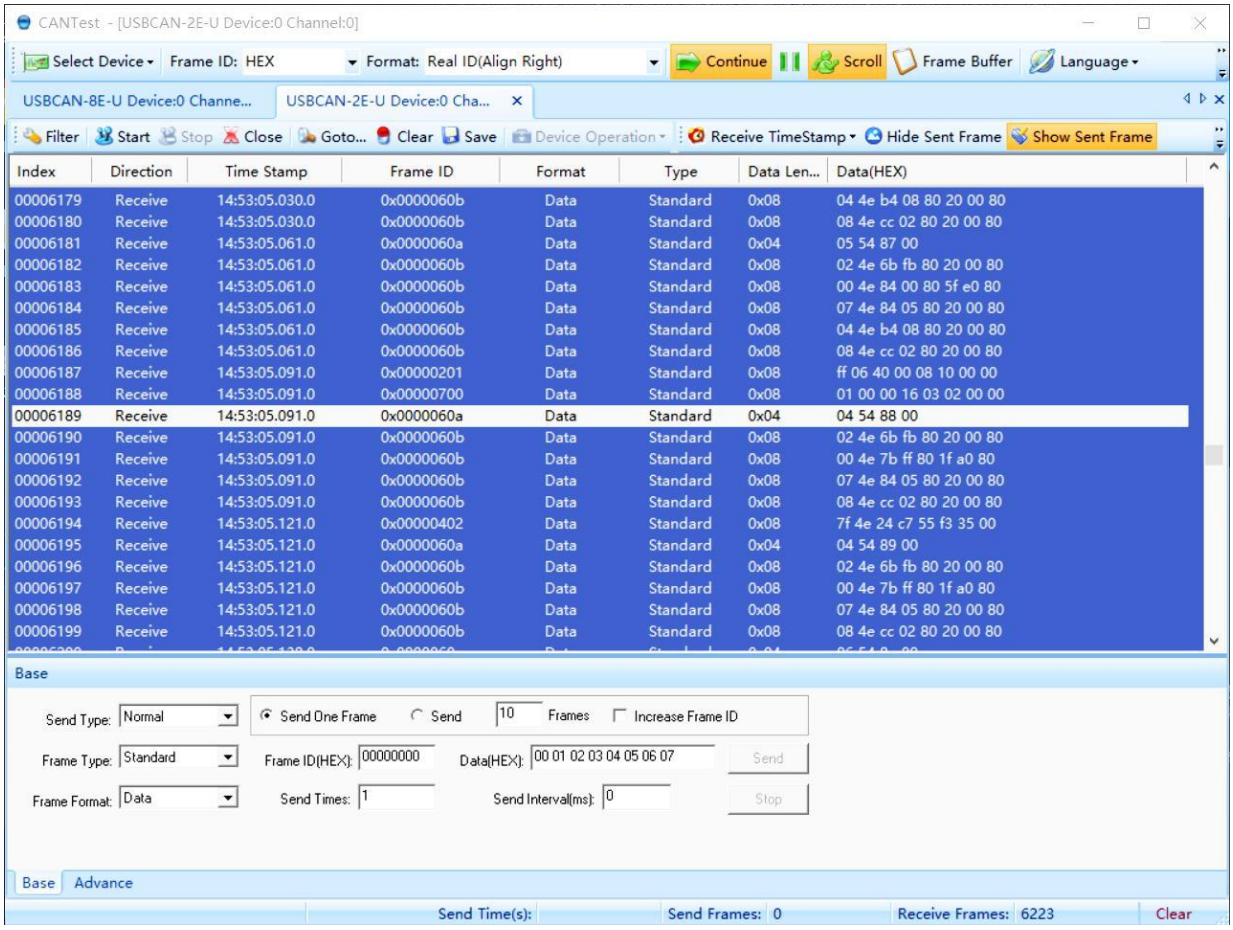


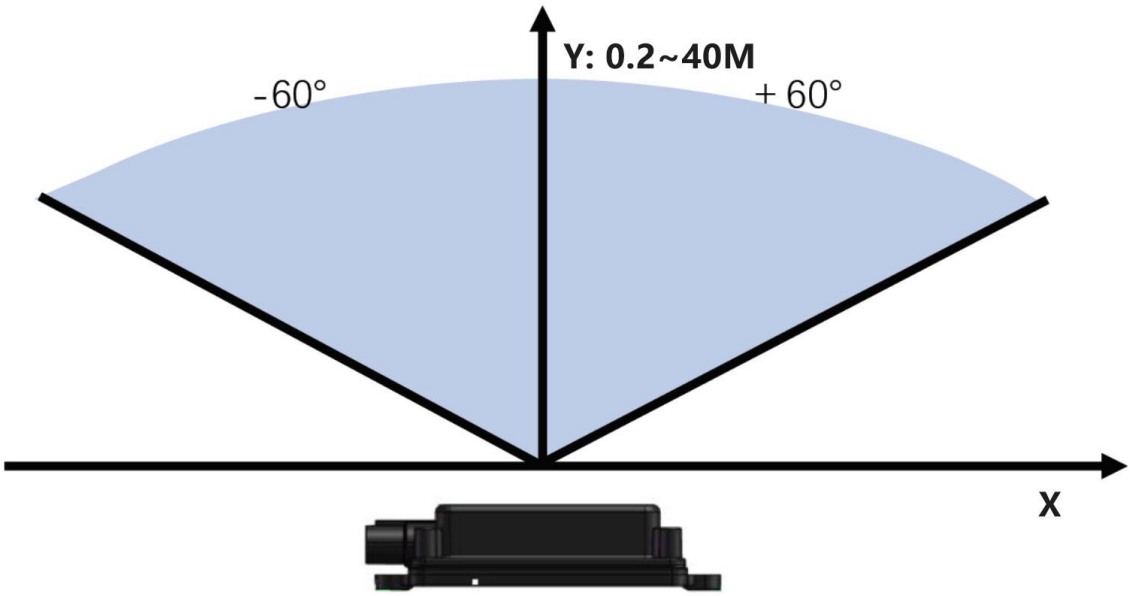
10. Windows Test Demo

10.1.: Radar_Tools.exe



10.2.: CANTest.exe





Radar coordinate diagram

Test Material List:

No.	Description	Qty
1	CAR-A60 77G Radar Sensor	1
2	Radar power/data wire	1
3	Windows OS PC	1
4	USB CANalyst-II tool	1
5	12VDC Power supplier	1
6	USB Camera(optional)	1
7	CAN test software	1

Any question, please feel free contact: info@lintechtt.com