MR72 Millimeter Wave Radar

Communication Protocol

Hunan Nanoradar Science and Technology Co.,Ltd.

Version History

Date	Version No.	Description
2018-12-17	1.0	the 1st version of MMV radar 77GHz MR72
2019-05-29	1.1	To optimize the table
2019-06-06	1.2	Add a parsing example
2020-03-02	1.3	Add CAN interface collision detection region drawing function

Content

1 CAN Interface Introduction

MR72 radar support CAN interface, the CAN bus communication network conforms to iso11888-2 standard, and the transmission rate is 500K bits/second. MR72 transmits radar signals to surrounding areas, and receives signals through multi-step processing, which can obtain track information of target group.

The introduction of radar MR72 UART interface protocol is on chapter 7.

Through a simple software interface, the sensor is connected to the CAN network to provide radar-based environmental awareness information to one or several evaluation units. Sensors can also be configured through the CAN interface.

The CAN interface of MR72 is used to configure sensors, to output sensor state information, input and output of sensor data. A CAN bus is able to mount up to 8 pieces MR72 sensors. The sensor ID can be configured, and the output Message IDs (Message IDs) can change.

If the sensor ID is 0 ~7, then the message ID can be calculated as follows:

MsgID= MsgID_0+SensorID*0x10

For example, if sensor ID=0, then the configuration message ID is 0x200; If sensorID=1, configuration message 0x210, and so on. When the sensor ID is set, the sensor will only respond to the new configuration message.

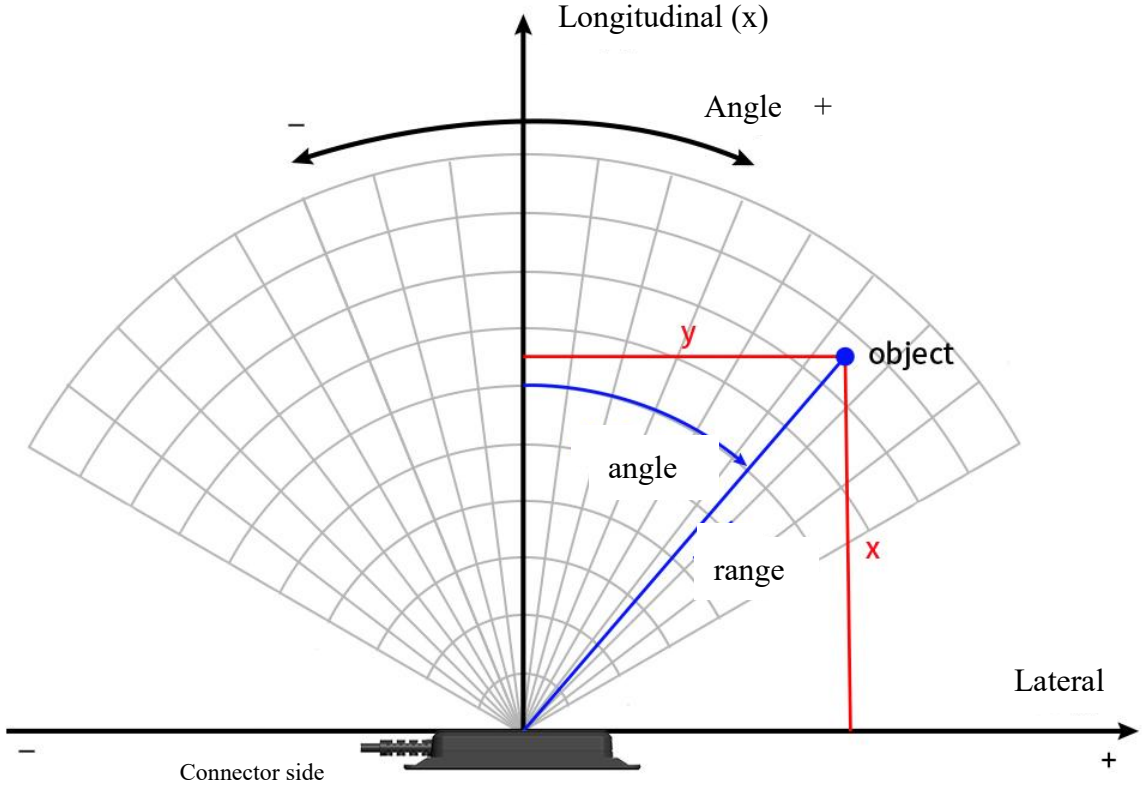
Radar Sensor CAN message (Sensor ID 0)

In/Out	ID	Message Name	Content	Section
In	0x200	RadarCfg	Radar configuration information	4.1
In	0x401	Region configuration	Collision region configuration	4.2
Out	0x201	RadarState	Radar statue information	5.1
Out	0x700	SoftWareVersion	Including software version and algorithm version	5.2
Out	0x402	Region State	Collision region sets statue	5.3
Out	0x60A	Obj_0_Status	Object statue information	6.1
Out	0x60B	Obj_1_General	Object general information	6.2

Where 0x201 and 0x700 are heartbeat signals, which are output once per second.

2 Radar Description

The MR72 USES 77GHz high-frequency electromagnetic waves to analyze the surrounding environment. The reflected signal is output as an object after several steps of processing. Objects are information about the location, speed and signal strength of radar reflections that are calculated and output once every period.
The Cartesian coordinates of the radar as below picture:

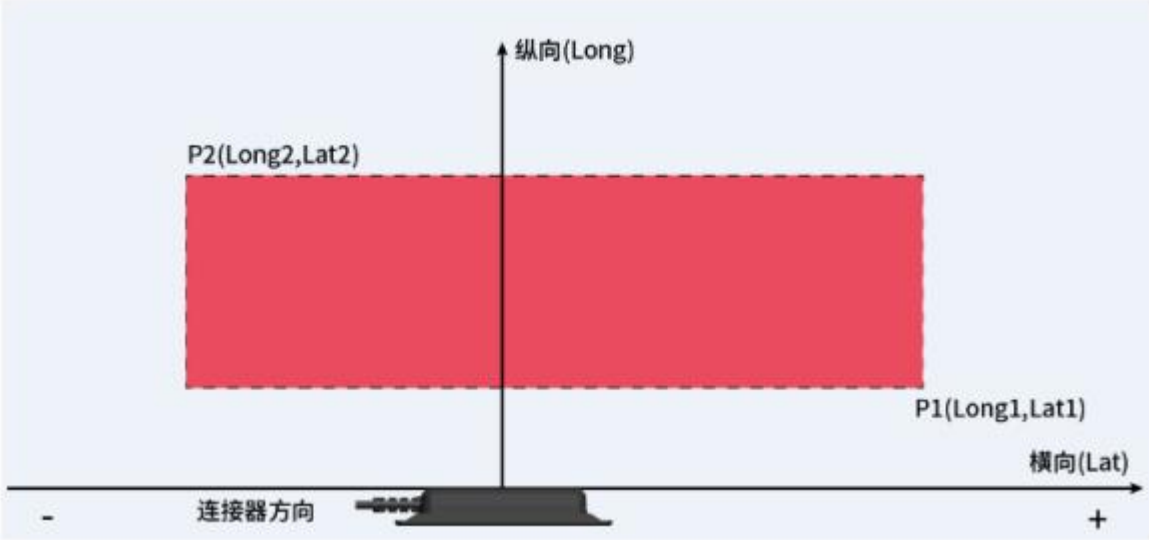


Where, the negative direction of CAN interface remains unchanged.
The default power of MR72 is 2W, and the peak power is 2.5w.
Be sure to use 5~32V DC power supply when working. The current is greater than 0.2A at 12V and 0.5A at 5V.
When the current or voltage is too low, the radar will not work properly.
As the target approaches, the velocity is negative (-). As the target leaving, the velocity is positive (+).

3 Collision Detection Region Sets Description

To meet customer's requirement, now add radar collision detection region sets function based on MR72 standard CAN protocol, the program outputs a rectangle region(±3 x 50m), objects are output in radial distance order from near to far.

The rectangle region sets as below:



Now support set only one input rectangle, two coordinates points P1, P2 determine a rectangle region, and to set detected objects output numbers on this region, allow output number is “N”, if “N” is greater than the number of actual detected objects in the rectangle region, then the actual number is output; If “N” is less than the number of actual detected objects in the rectangle region, then the “N” is output.(Output in order of distance near to far).

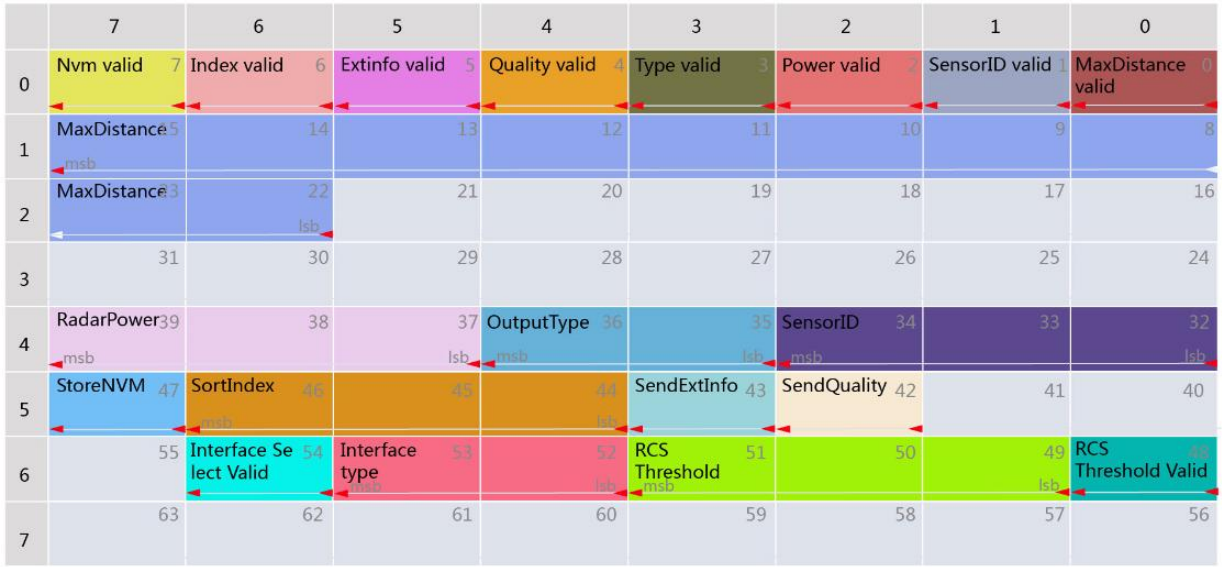
Where, “N” MAX is 63. P1(Long1, Lat1), Long1 is Vertical, Lat1 is Abscissa.

4 Configuration Input

4.1 Parameter Configuration

The basic configuration parameters of the radar sensor can be set with message RadarCfg 0x200. It is not necessary to cyclically repeat the configuration message. To store the configuration in the non-volatile memory (NVM) to be automatically set at start up on any subsequent power up, the signal RadarCfg_StoreInNVM has to be set to active (0x1). It is important to note that the number of writes to the NVM should be kept to a minimum as this could reduce the service life of the memory.

0x200 can be modified several or with just one configuration parameter. For each argument, message contains one significant bit. If the valid bit is set to 1, the corresponding change is Valid; otherwise, the corresponding change is invalid.



Signal	Start	Len	Min	Max	Res	Uint
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	40-80m
RadarCfg_SensorID	32	3	0	7	1	ID(0~7)

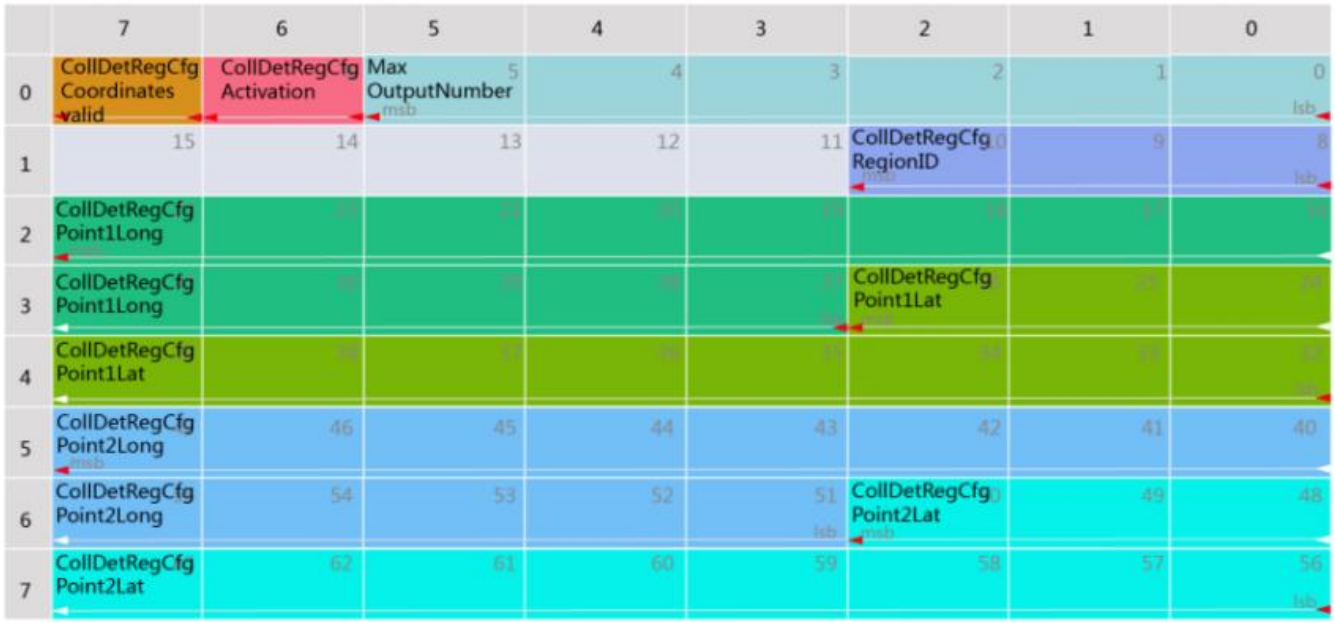
RadarCfg_OutputType	35	2	0	2	1	0x0:none 0x1:Objects 0x2:Clusters
RadarCfg_RadarPower	37	3	0	7	1	0x0:Standard 0x1:-3dB Tx gain 0x2: -6dB Tx gain 0x3: -9dB Tx gain
RadarCfg_SortIndex	44	3	0	7	1	0x0:No Sorting 0x1: Sorted by range 0x2: Sorted by RCS
RadarCfg_StoreNVM	47	1	0	1	1	0x0:Inactive 0x1: Active
RadarCfg_RCS_Threshold_Valid	48	1	0	1	1	0x0:Invalid 0x1:Valid
RadarCfg_RCS_Threshold	49	3	0	7	1	0x0:Standard 0x1:High Sensitivity

Remarks：

1. In multi-target mode, the target is output in order of distance from near to far.
2. For example, when modifying the radar ID, RadarCfg_SensorID_Valid is enabled first, and RadarCfg_SensorID is the ID to be set, which is successfully modified at present. If you need to save a change to NVM, and the next time you start it in effect, you need to enable the RadarCfg_StoreInNvm_Valid bit.
3. MR72 only supports Object mode, not Cluster mode.
4. Res stands for resolution. For example, if the maximum distance value is configured as 20 and the resolution is 2, then the maximum distance value is 40 meters.

4.2 0x401 Instruction for Rectangle Region and Object Number Sets

Collision detection region configuration(0x401) now support set only one rectangle region, the sets instruction format as below:



Signal	Start	Len	Offset	Min	Max	Res	Description
Max_OutputNumber	0	6		0	63	1	The allowed MAX output object numbers
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	Region ID number, default value is "1"
CollDetRegCfg_Point1Long	27	13	-500	-500	1138.2	0.2	Unit is "meter"
CollDetRegCfg_Point1Lat	32	11	-204.6	-204.6	204.8	0.2	Unit is "meter"
CollDetRegCfg_Point2Long	51	13	-500	-500	1138.2	0.2	Unit is "meter"
CollDetRegCfg_Point2Lat	56	11	-204.6	-204.6	204.8	0.2	Unit is "meter"

Max_OutputNumber: the allowed output MAX object number "N", where the "N" <64. If "N" is greater than the number of actual detected objects in the rectangle region, then the actual number is output; If "N" is less than the number of actual detected objects in the rectangle region, then the "N" is output.(Output in order of distance near to far).

CollDetRegCfg_Activation: It is collision detection region enabled, where "0x0:inactive" means not enabled; "0x0:active" means enabled, after enabled this function and your coordinates points P1, P2 is valid then your rectangle region sets could be set valid. Be remember to power off radar and to valid the last time rectangle region sets.

CollDetRegCfg_CoordinatesValid: It's the coordinates sets, where "0x0:invalid" means coordinates sets not valid; "0x0:valid" means coordinates sets valid. Be sure to enable "**CollDetRegCfg"** and to assure the enable coordinates is valid then the coordinates sets could be valid.

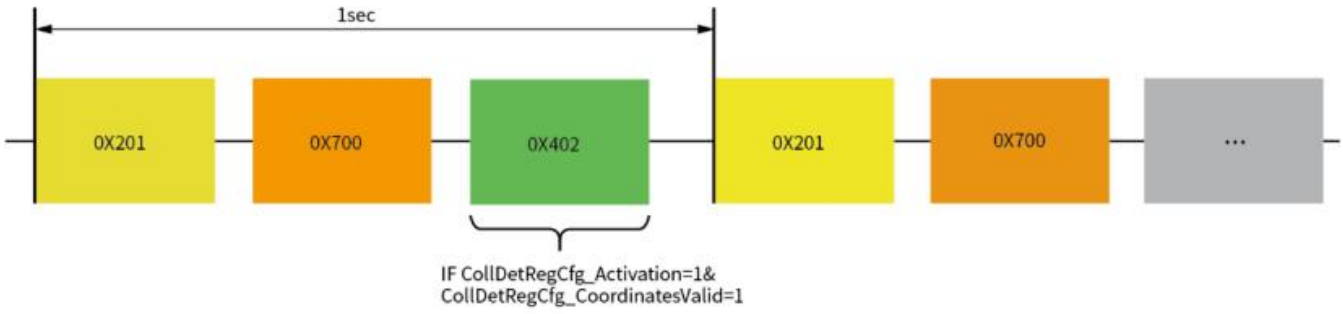
CollDetRegCfg_Point1Long: The longitudinal distance of coordinate point 1.
CollDetRegCfg_Point1Lat: The transverse distance of coordinate point 1.
CollDetRegCfg_Point2Long: The longitudinal distance of coordinate point 2.
CollDetRegCfg_Point2Lat: The transverse distance of coordinate point 2.
Attention when set the coordinate points: Point 1 is the value at the button right corner of the rectangle region; Point 2 is the value at the upper left corner of the rectangle region.
Assure:
CollDetRegCfg_Point1Long < CollDetRegCfg_Point2Long
CollDetRegCfg_Point1Lat > CollDetRegCfg_Point2Lat
Otherwise the sets would be invalid and will not be saved at flash.

5 State Output

The sensor is always sending cyclically (once per second) the current configuration and sensor state in message 0x201 and current firmware version in message 0x700.

If Enable collision detection region and the coordinate sets is valid, then output 0x402.

If collision detection region or coordinates sets is invalid, then output objects modes data, no 0x402 output.



Periodically send status information

5.1 Radar Statue Information(0x201)

The message RadarState (0x201) is sent by the sensor in a regular interval (once per second). After configuring a radar configuration parameter, the corresponding signal in message 0x201 can be checked in order to verify that the configuration change has been accepted.

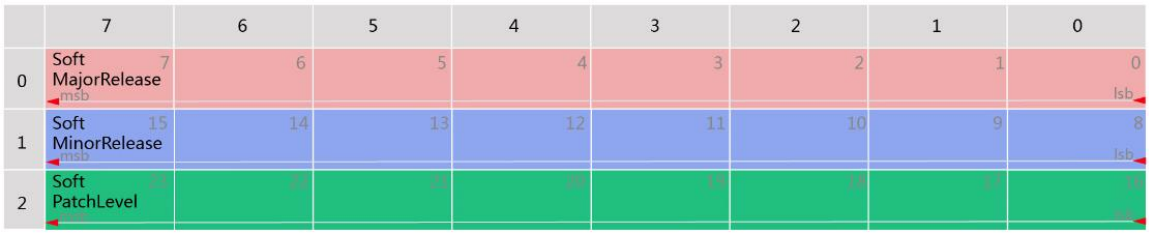
Signal	Start	Len	Min	Max	Res	Uint
RadarState_NVMReadStatus	6	1	0	1	1	0x0:failed 0x1:Successful
RadarState_ NVMWriteStatus	7	1	0	1	1	0x0:failed 0x1:Successful
RadarState_ MaxDistanceCfg	22	10	0	2046	2	m(40-160m are supported in standard version)
RadarState_SensorID	32	3	0	7	1	Current radar ID(0~7)
RadarState_SortIndex	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 0x1:-3dB Tx Gain 0x2: -6dB Tx Gain 0x3: -9dB Tx Gain
RadarState_OutputTypeCfg	42	2	0	3	1	0x0:none 0x1:Objects 0x2:Clusters
RadarState_RCS_threshold	58	3	0	7	1	0x0:Standard

						0x1:high sensitivity

Remark：

1. MR72 not support Cluster mode.

5.2 Version Information(0x700)



Signal	Start	Len	Min	Max	Res	Description
Soft_MajorRelease	0	8	0	255	1	Software major release
Soft _MinorRelease	8	8	0	255	1	Software minor release
Soft _PatchLevel	16	8	0	255	1	Software patch level

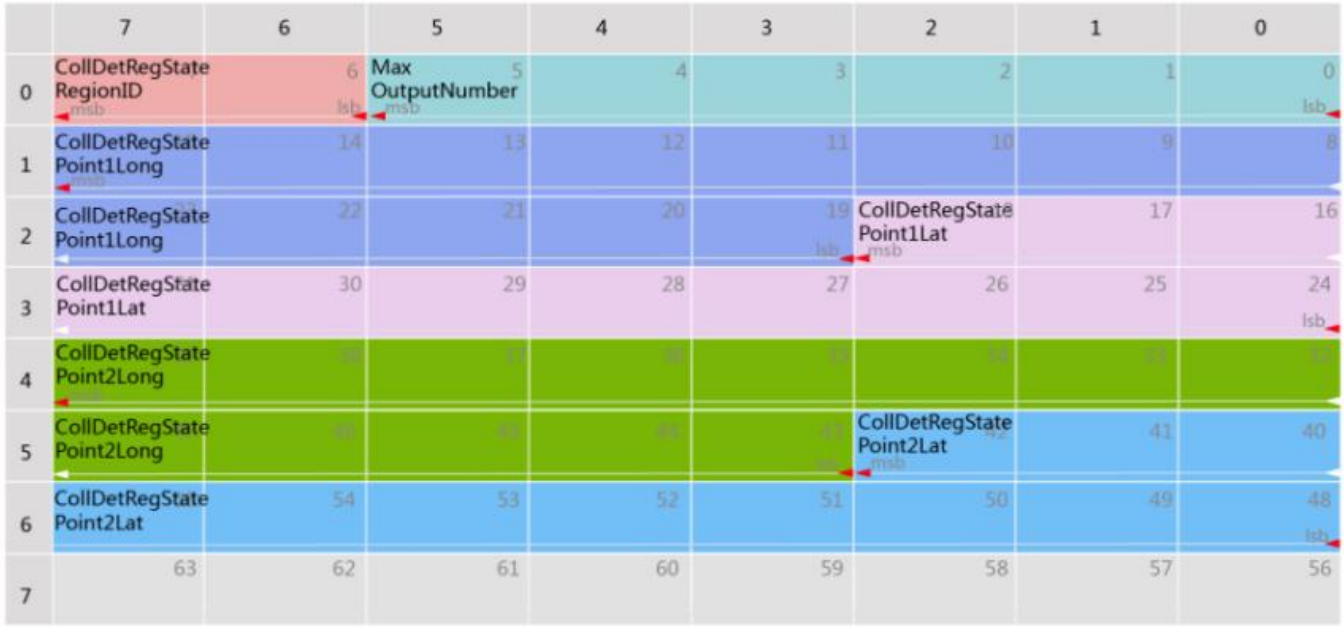
Soft_MajorRelease: Software major release

Soft_MinorRelease: Software minor release

Soft_PatchLevel: Software patch level

5.3 Collision Detection Region Sets Statue(0x402)

When 0x401: CollDetRegCfg_Activation = 1 and CollDetRegCfg_CoordinatesValid =1, radar send collision detection statue information periodically.
When 0x401: CollDetRegCfg_Activation or CollDetRegCfg_CoordinatesValid value is not “1”, radar output object mode data.

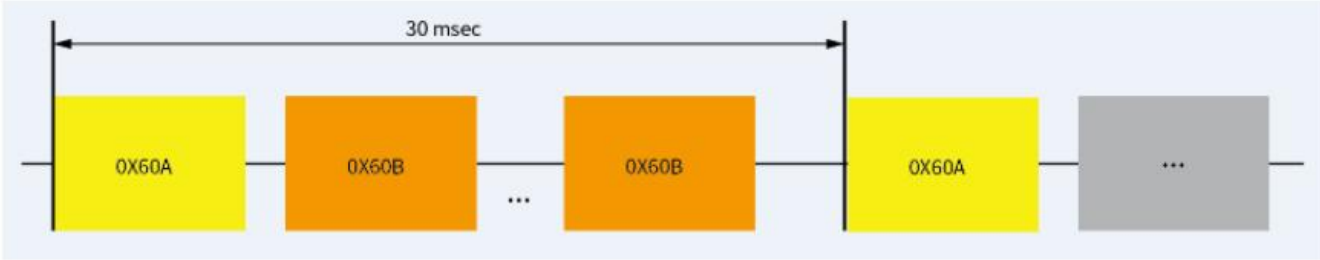


Signal	Start	Len	Offset	Min	Max	Res	Description
Max_OutputNumber	0	6		0	63	1	The allowed MAX numbers
CollDetRegState_RegionID	6	2		0	7	1	Region ID, default is 1
CollDetRegState_Point1Long	19	13	-500	-500	1138.2	0.2	Unit is meter
CollDetRegState_Point1Lat	24	11	-204.6	-204.6	204.8	0.2	Unit is meter
CollDetRegState_Point2Long	43	13	-500	-500	1138.2	0.2	Unit is meter
CollDetRegState_Point2Lat	48	11	-204.6	-204.6	204.8	0.2	Unit is meter

6 Object list

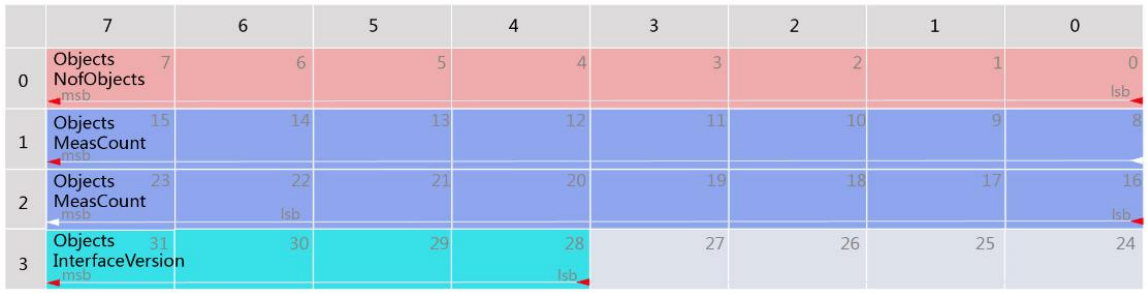
Object list is including 2 messages, Object_0_Status(0x60A) is the first list header message, which is contains information about the number of targets to be sent.

Object_1_General (0x60B) is the second message, this message contains information about the speed and distance of the target. Periodically send all tracking target information.



6.1 Object List Status(0x60A)

Object List Status(0x60A) is contains the target list header message, first sending 0x60A message during each measurement cycle.



Signal	Start	Len	Min	Max	Res	Description
Objects_NofObjects	0	8	0	255	1	Number of targets detected
Objects_MeasCount	8	16	0	65535	1	Cycle count
Objects _InterfaceVersion	28	4	0	15	1	Target list CAN interface version, default is 0

6.2 Objects General Information(0x60B)

This message contains the location and speed of the target, periodically sending all tracking target information.

Signal	Start	Len	Min	Max	Res	Description
Objects_ID	0	8	0	255	1	Target ID
Objects _DistLong	19	13	-500	1138.2	0.2	m
Objects _Distlat	24	11	-204.6	+204.8	0.2	m
Objects _VrelLong	46	10	-128.00	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
Sector_Number	51	2	1	3	1	0x1:1Sector 0x2: 2Sector 0x3: 3Sector

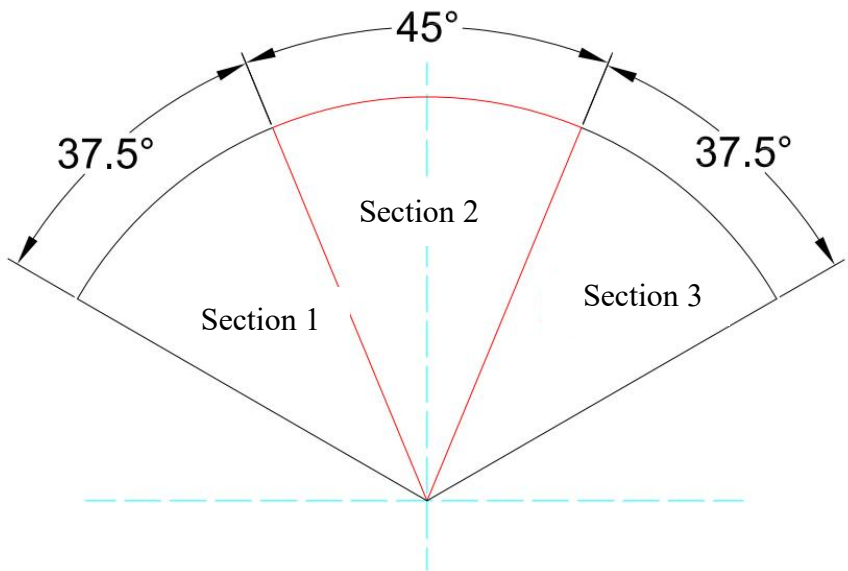
Objects _ VrelLat	53	9	-64	63.75	0.25	m/s
Objects _RCS	56	8	-64	63.5	0.5	dBm2

Objects_ID： Target ID；
Objects _DistLong： Target radial distance；
Objects _Distlat： Target lateral distance；
Objects _ VrelLong： Target radial speed;
Objects _DynProp： Target dynamic properties, now all default values are 0
Objects _ VrelLat： Target lateral velocity；
Objects _RCS： Target RCS，default are 0.

Remarks：

1. Now Objects _DynProp are 0.

2. Sector_Number represents the sector where the target is located, and the sector division is shown in the figure below:



3. If the version is three-sector, it will only output the nearest target of the three-sector. The output order is sector 1, sector 2, and sector 3. If the sector has no target, the distance velocity value will be assigned to 0.

4. If the version is multi-target, the nearest target of three sectors will be output first, and the output order is sector 1, sector 2, sector 3. If no target output, and then the remaining target will be output in the distance order (from near to far).

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

6. Min:Offset. Some values may have negative values (-). In order to ensure output positive

value (+) and for convenient analysis, the radar end shall increase the offset. When analyzing radar data, offset is needed.

If The radar data is Value1, then the actual value is Value2 = Value1*Res+Min.

7 Example for CAN protocol parsing

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

Min: Offset. Some values may have negative values (-). In order to ensure output positive value (+) and for convenient analysis, the radar end shall increase the offset. When analyzing radar data, offset is needed.

If The radar data is Value1, then the actual value is Value2 = Value1*Res+Min.

7.1 Configuration message example

The default program type of “radar_outputType” is “Objects”, it will be revised and saved to “NVM”;

“radar_power” is “standard”;

SortIndex is classified as distance (range);

RCS_Threshold is “standard”;

SensorID is “0”;

MaxDistance is 160 meters.

Example: if ID is 0*200, then the message content is 0xFF 0x14 0x00 0x00 0x09 0x90 0x00 0x00

If this message modify the radar ID to 1, and save the modified parameters to “NVM”.

The next time the radar is powered on, ID is 1. The max distance is 160 meters, that is (0x14 x 4 + 0x00>>6) x 2; the value resolution is 2, so 0x50 = 80, 80 times

2, then the max distance is 160meters.

When configure the bit, bit 0 needs to enable byte 0, and enable maxdistance_valid to 1; If bit 0 maxdistance_valid of byte 0 is “0”, then the configuration is invalid.

When the change is started, the valid bit must be valid for the configuration to take effect, otherwise the configuration doesn’t take effect.

7.2 MR72 Common configuration commands（CANMonitor）

The CAN interface of MR72 is used to configure sensors, to output sensor state information, to input and output of sensor data.

A CAN bus is able to mount up to 8pcs MR72 sensors. The sensor ID can be configured, and the output Message IDs (Message IDs) can change.

The ID of sensor is 0 ~7, The message ID is calculated as follows:

MsgID= MsgID_0+SensorID*0x10

Ignore the message ID and focus on the message content as below.

1. If modify the radar ID to 1 and save it, the command is as below.
   82 00 00 00 01 80 00 00
2. If modify the radar ID to 2 and save it, the command is as below
   82 00 00 00 02 80 00 00
3. If modify the radar ID to 3 and save it, the command is as below
   82 00 00 00 03 80 00 00
4. If modify radar to high sensitivity(to detect fine targets), the command is as below
   80 00 00 00 00 80 03 00
5. If modify radar to low sensitivity, the command is as below
   80 00 00 00 00 80 01 00
6. If modify radar to CAN interface output, the command is as below
   80 00 00 00 00 80 40 00
7. If modify radar to UART interface output, the command is as below
   80 00 00 00 00 80 50 00

MR72 Common configuration commands（NSM PC software）

Kindly refers to NSM PC software application manual for details.

7.3 60B Parsing Example

As shown above, the current radar id is 5.

Message is 0x57 0x4E 0xC4 0x0C 0x7F 0x60 0x18 0x80

Which is：

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x57	0x4E	0xC4	0x0C	0x7F	0x60	0x18	0x80

1. Object ID：0x57=87，which means the object ID is 87. The target is generated in a loop between 0 and 255. In the stable tracking process, the target ID remains unchanged;
2. Object radial distance：(0x4E*32 +0xC4>>3)*0.2-500 = 4 metes；
3. Object lateral range：((0xC4 &0x07)*256 + 0x0C) *0.2-204.6 = 2.6 meters；
4. Object radial speed：(0x7F*4+0x60>>6)*0.25-128 = -0.75m/s；
5. Object lateral speed：((0x60&0x3F)*8+0x18>>5)*0.25-64 = 0m/s；
6. Object dynamic attribute：0x18&0x07 = 0, the default value is 0；
7. Sector No.: (0x18>>3)&0x03 = 3, which means the target is in the three sector;
8. RCS：0x80 *0.5-64= 0，the default value is 0.

7.4 Heartbeat Signal（0x700）

The first three bytes of the message represent the software version number. If the return message is 0x700, which means the current radar ID is 0; If returns 0x710, means the current radar ID is 1.

Example: the return message is 0x01 0x00 0x15 0x00 0x00 0x00 0x00 0x00

The the radar version is V 1.0.21.

7.5 Command for Collision Detection Region Configuration

(1) The way to set rectangle region as 60x20 meters, the MAX output object is 63, enable region set and valid coordinate points. If point1 coordinate is (0,3); point2 coordinate is (20,-3), then the 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 detail phrasing as below:

	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	0	1	1	1	0	0x0E
5	0	1	0	1	0	0	0	1	0x51
6	0	1	0	0	0	0	1	1	0x43
7	1	1	1	1	0	0	0	0	0xF0

(2) The way to set rectangle region as 10x50 meters, the MAX output object is 63, enable region set and valid coordinate points. If point1 coordinate is (0,5); point2 coordinate is (50,-5), then the 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 detail phrasing as below:

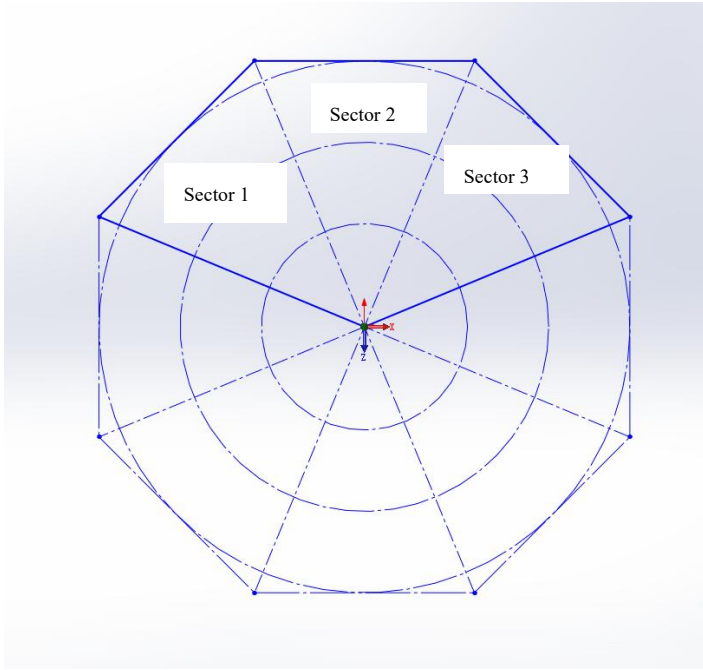
	7	6	5	4	3	2	1	0	Hex
0	1	1	1	1	1	1	1	1	0xFF
1	0	0	0	0	0	0	0	1	0x01
2	0	1	0	0	1	1	1	0	0x4E
3	0	0	1	0	0	1	0	0	0x24
4	0	0	0	1	1	0	0	0	0x18
5	0	1	0	1	0	1	0	1	0x55
6	1	1	1	1	0	0	1	1	0xF3
7	1	1	1	0	0	1	1	0	0xE6

Attention when set collision detection region:
(1) Assure (point1Long < point2Long) & (point1Lat > 18 point2Lat). Otherwise the sets won't be saved.
(2) As long as the coordinates meet the conditions, save the parameters to FLASH regardless of whether the activated locale and coordinates are valid;
(3) But only if the locale and coordinate are both valid, the rectangle box will be drawn, and 0x402 will be output periodically.
(4) Region settings and coordinates as long as one of them is invalid, the radar enters the objects multi-target mode and outputs all the objects standard, that is, no rectangle region, no 0x402 output..

8 UART Protocol

The serial version of MR72 supports 115200 baud, 8-bit data bit 1 stop bit, and the sending cycle is 60ms.

The targets of the three sectors detected by UART output are divided into the following figure:



At present, MR72 only outputs the nearest target of sector 1, sector 2 and sector 3 within the detection range.

Among them, the beam width of the MR72 azimuth plane is ±56 °, and the beam width of the elevation plane is ±7 °

Data sending format (ASCII code) :

The head byte is D1 D2 D3 D4 D5 D6 D7 D8 CRC8

The data protocol format is shown in the following table：

	Bytes number	Instruction
Head byte	2Btyes	Fixed value，'T' 'H' T ascii for a decimal 84 (0x54) H ascii for a decimal 72(0x48)
D1	2Btyes	The nearest target of Sector 2, if data invalid, writes 0xFFFF, the high 8-bit byte comes first, and the low 8-bit byte comes last
D2	2Btyes	The nearest target of Sector 3, if data invalid, writes 0xFFFF, the high 8-bit byte comes first, and the low 8-bit byte comes last
D3	2Btyes	The obstacle distance of 90 degree sector, if data invalid, writes 0xFFFF, the high 8-bit byte comes first, and the low 8-bit byte comes last

D4	2Btyes	The obstacle distance of 135 degree sector, if data invalid, writes 0xFFFF, the high 8-bit byte comes first, and the low 8-bit byte comes last
D5	2Btyes	The obstacle distance of 180 degree sector, if data invalid, writes 0xFFFF, the high 8-bit byte comes first, and the low 8-bit byte comes last
D6	2Btyes	The obstacle distance of 225 degree sector, if data invalid, writes 0xFFFF, the high 8-bit byte comes first, and the low 8-bit byte comes last
D7	2Btyes	The obstacle distance of 270 degree sector, if data invalid, writes 0xFFFF, the high 8-bit byte comes first, and the low 8-bit byte comes last
D8	2Btyes	The nearest target of Sector 1, if data invalid, writes 0xFFFF, the high 8-bit byte comes first, and the low 8-bit byte comes last
CRC8	1Btyes	CRC8 check, the checking algorithm please see CRC8 checking program

CRC8 checking program

Crc.cpp

```
static const uint8_t crc8_table[] = {
0x00, 0x07, 0x0e, 0x09, 0x1c, 0x1b, 0x12, 0x15, 0x38, 0x3f, 0x36, 0x31,
0x24, 0x23, 0x2a, 0x2d, 0x70, 0x77, 0x7e, 0x79, 0x6c, 0x6b, 0x62, 0x65,
0x48, 0x4f, 0x46, 0x41, 0x54, 0x53, 0x5a, 0x5d, 0xe0, 0xe7, 0xee, 0xe9,
0xfc, 0xfb, 0xf2, 0xf5, 0xd8, 0xdf, 0xd6, 0xd1, 0xc4, 0xc3, 0xca, 0xcd,
0x90, 0x97, 0x9e, 0x99, 0x8c, 0x8b, 0x82, 0x85, 0xa8, 0xaf, 0xa6, 0xa1,
0xb4, 0xb3, 0xba, 0xbd, 0xc7, 0xc0, 0xc9, 0xce, 0xdb, 0xdc, 0xd5, 0xd2,
0xff, 0xf8, 0xf1, 0xf6, 0xe3, 0xe4, 0xed, 0xea, 0xb7, 0xb0, 0xb9, 0xbe,
0xab, 0xac, 0xa5, 0xa2, 0x8f, 0x88, 0x81, 0x86, 0x93, 0x94, 0x9d, 0x9a,
0x27, 0x20, 0x29, 0x2e, 0x3b, 0x3c, 0x35, 0x32, 0x1f, 0x18, 0x11, 0x16,
```

```
        0x03, 0x04, 0x0d, 0x0a, 0x57, 0x50, 0x59, 0x5e, 0x4b, 0x4c, 0x45, 0x42,
        0x6f, 0x68, 0x61, 0x66, 0x73, 0x74, 0x7d, 0x7a, 0x89, 0x8e, 0x87, 0x80,
        0x95, 0x92, 0x9b, 0x9c, 0xb1, 0xb6, 0xbf, 0xb8, 0xad, 0xaa, 0xa3, 0xa4,
        0xf9, 0xfe, 0xf7, 0xf0, 0xe5, 0xe2, 0xeb, 0xec, 0xc1, 0xc6, 0xcf, 0xc8,
        0xdd, 0xda, 0xd3, 0xd4, 0x69, 0x6e, 0x67, 0x60, 0x75, 0x72, 0x7b, 0x7c,
        0x51, 0x56, 0x5f, 0x58, 0x4d, 0x4a, 0x43, 0x44, 0x19, 0x1e, 0x17, 0x10,
        0x05, 0x02, 0x0b, 0x0c, 0x21, 0x26, 0x2f, 0x28, 0x3d, 0x3a, 0x33, 0x34,
        0x4e, 0x49, 0x40, 0x47, 0x52, 0x55, 0x5c, 0x5b, 0x76, 0x71, 0x78, 0x7f,
        0x6a, 0x6d, 0x64, 0x63, 0x3e, 0x39, 0x30, 0x37, 0x22, 0x25, 0x2c, 0x2b,
        0x06, 0x01, 0x08, 0x0f, 0x1a, 0x1d, 0x14, 0x13, 0xae, 0xa9, 0xa0, 0xa7,
        0xb2, 0xb5, 0xbc, 0xbb, 0x96, 0x91, 0x98, 0x9f, 0x8a, 0x8d, 0x84, 0x83,
        0xde, 0xd9, 0xd0, 0xd7, 0xc2, 0xc5, 0xcc, 0xcb, 0xe6, 0xe1, 0xe8, 0xef,
        0xfa, 0xfd, 0xf4, 0xf3
};

/*
   crc8 from trone driver by Luis Rodrigues
 */
uint8_t crc_crc8(const uint8_t *p, uint8_t len)
{
 uint16_t i;
 uint16_t crc = 0x0;

 while (len--) {
        i = (crc ^ *p++) & 0xFF;
        crc = (crc8_table[i] ^ (crc << 8)) & 0xFF;
 }

 return crc & 0xFF;
}
```

Call way: crc8 = crc_crc8(buffer, 18)；//buffer is receiving the cached array for the data.