LXSDF T2 COMMUNICATION STANDARD

LX Serial Data Format Type 2 stream and data simultaneously transmitted General-Purpose Serial Communication Standard.

Doc. ID. LXD12 V1.1

Release Date. 2012-12-03 .

Abstract - LXSDF T2 (LX Serial Data Format Type2) is General-purpose Serial Communication Format, which it is able to transmit real-time stream data and low speed data relatively in one packet. A typical example of Stream data is the time series data converted AD of analog signal. This manual illustrates RX data format of LXSDF T2 and TX Packet Structure. Also, it indicates receiving and processing code examples about TX Packet from host, the way to search Com port connected with equipment automatically and code example.

CONTENTS

PICTURE CONTENTS.

TABLE CONTENTS.

LXSDF T2 ABSTRACT.

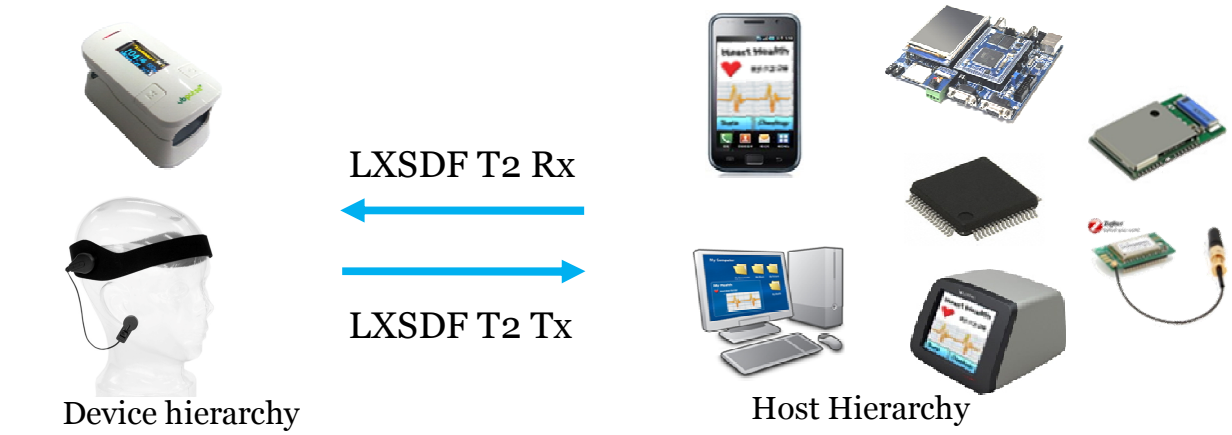
LXSDF T2 (LX Serial Data Format Type2) is General-purpose Serial Communication Format, which is able to transmit the real-time stream data and the low speed data relatively in one packet. Stream data means data related in time like time series data converted AD of analog signal. Serial Communication indicates COM port, serial port and Rs232 as if these are indicated as UART (Universal Asynchronous Receiver & Transmitter), PC and Smart phone etc.. in embedded field.

Serial Communication Standard is the type that is transmitted in one byte sequentially, but it rarely happens that sends only one byte of data element needed in Communication in real applied process. For example, when transmitting sample data converted into 12 bit AD, must transmit serial communication 2 bytes dividing 4 bits and 8 bits and unify them into one data in receiver. If there are many kinds of data to transmit, packet notion is asked. Standard which can handle several bytes as one byte is needed, transmitting and receiving sides use the data by separating according to the standard.

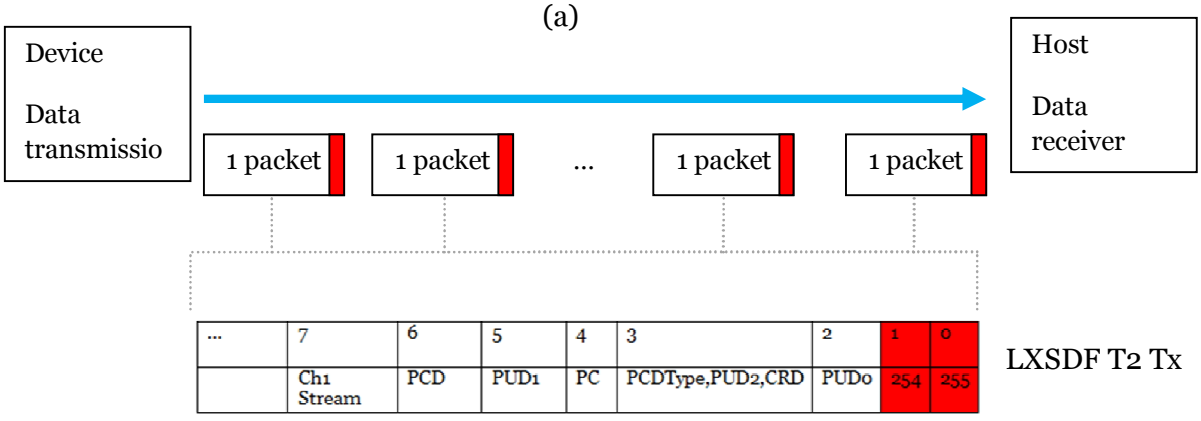
The Standard of Packet Format can be various for using and it can transmit the real-time multi channel wave form data at high speed. Also, low speed data like state information of device and calculation value are suitable serial communication format for all transmittable general purpose real-time communication application as the same packet type.

LXSDF T2 Rx and Tx CLASSIFICATION.

LXSDF T2 is defined as 2 kinds (Rx , Tx) of data format. In this context, Rx and Tx mean reception and transmission respectively and the direction of communication as the device communicated with host. The left side is device example and the right is host as the following picture. There are general PC, Smart phone and general embedded system etc., as host. Normal devices have many cases that they transmit a lot of data to the host side, while the host side has the cases that they usually transmit simple orders. The data type transmitting from device to host (receiving from device as host) is named LXSDF T2 Packet and the data type receiving from host to device is named LXSDF T2 Rx.



(a)



(b)

Fig 1. LXSDF T2 Rx, Tx Classification, Detail Structure of Tx Packet

LXSDF T2 Rx DATA FORMAT

This is the data standard when transmitting data from host to device.

It uses sequential 3 bites and transmits the data from host in order of Rx Index 1,2,3. The most significant bit of RX Index which transmits the first in time is 1, the most significant bit of the rest has to be recorded as 0 definitely. Those are defined differently in device as how to work by products according to Cmd data and refer to Communication Standard of the products.

Rx Index	name	Standard
0	Cmd 0	Most significant bit = 1
1	Cmd 1	Most significant bit = 0
2	Cmd 2	Most significant bit = 0

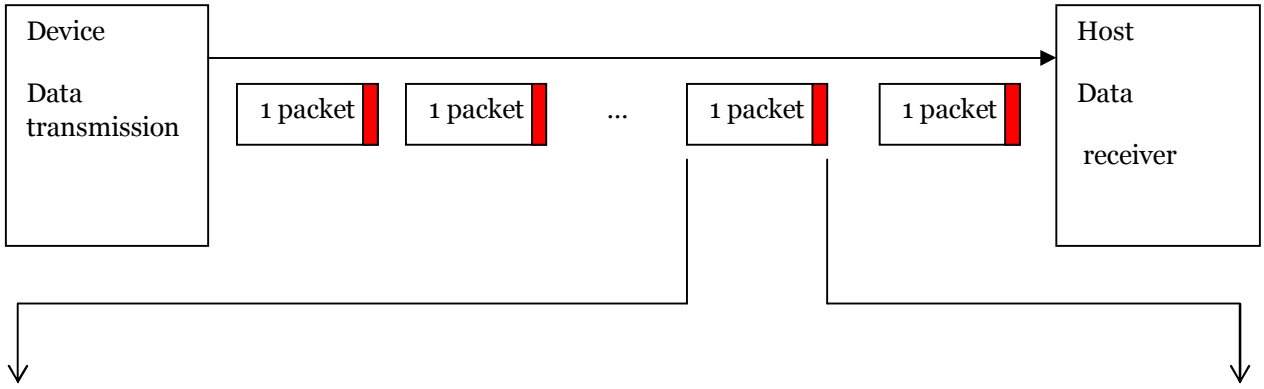
TABLE 1. LXSDF T2 Rx Format

LXSDF T2 TX PACKET

LXSDF T2 is defined differently between the data standard of transmitting data from device to host like a smart phone (Transmission in device, TX) and the data standard of transmitting data from host to device (receiver in device , RX)

RX data Format is very simple because data contents transmitted from host to device is not asked to transmit large bytes while TX data Format transmitting data from device to hosts is able to transmit various information when it can handle series of scores ~ hundreds bytes in 1 packet. The host side receiving the data from device needs the method to detect the starting spot of packet. LXSDF T2 TX packet uses initial 2 bytes for each packet transmission as Sync byte and records 255 for the first 1 byte and 254 for the next byte sequentially. Namely, Sync byte is designed only in the spots where appear 255 and 254 sequentially in the total packet byte arrays. When transmitting the data in device, it always transmits the data abiding by Standard. Receiving side monitors each byte and detects Sync byte, so it can find the starting spot of 1 packet. Once finding the starting spot, it is able to extract the data needed in the program under the TX packet Standard.

The following picture indicates that the red part in the first 1 packet is assigned 2 sync bytes and then a series of byte unit data every time.



...	7	6	5	4	3	2	1	0
	Ch1 Stream	PCD	PUD1	PC	PCDType,PUD2,CRD	PUD0	254	255

LXSDF T2 Tx

LXSDF T2 TX PACKET FORMAT.

Each index is one byte and transmitted in numerical order when transmitting serial.

TX INDEX	PACKET ELEMENT	EXPLANATION	VALUE
0	Sync byte 0	Sync byte 0	**255**
1	Sync byte 1	Sync byte 1	**254**
2	**PUD 0** (Packet Unit Data 1)	Bit7 ~Bit0 : General data.	0~255
3	**CRD** (Command Response Data)	Bit6. Reverse status bit in case device receive the RX data from host.	Bit7=0
	PUD 2 (Packet Unit Data 2)	Bit5,4,3 : general data	
	PCD Type (Packet Cyclic Data Type,)	Bit2,1,0 : If the data value is different, it means packet cyclic data type is different.	
4	**PC** (Packet Count)	Increase 1 for transmission every time. Start 0 again after the maximum value	0~255. Change Max on PCD Type
5	**PUD 1** (Packet Unit Data 0)	Bit6~Bit0 : General data	Bit7 = 0
6	**PCD** (Packet Cyclic Data)	PCD Type is allotted in PCD Packet Count is used as PCD identification (Note 1)	0~255
7	Ch1 Stream high	Bit6,5,4 : General data Low 4 bits: Ch1 Stream high 4	Bit7= 0

8	Ch1 Stream low	Ch1 Stream low 8	0~255
9	Ch2 Stream high	Bit6,5,4 : General data Low 4 bits: Ch2 Stream high 4	Bit7 = 0
10	Ch2 Stream low	Ch2 stream low 8	0~255
11	Ch3 Stream high	Bit6,5,4 : General data Low 4 bits: Ch3 Stream high 4	Bit7 = 0
12	Ch3 Stream low	Ch3 Stream low 8	0~255
13	Ch4 Stream high	Bit6,5,4 : General data Ch4 Stream high 4	Bit7 = 0
14	Ch4 Stream low	Ch4 Stream low 8	0~255
15	Ch1 Stream high	Bit6,5,4 : General data Ch1 Stream high 4	Bit7 = 0
16	Ch1 Stream low	Ch1 Stream low 8	0~255
..			
70 (Note 2)	Ch4 Stream low		0~255

Table 2. LXSDF T2 Tx Format

Note 1

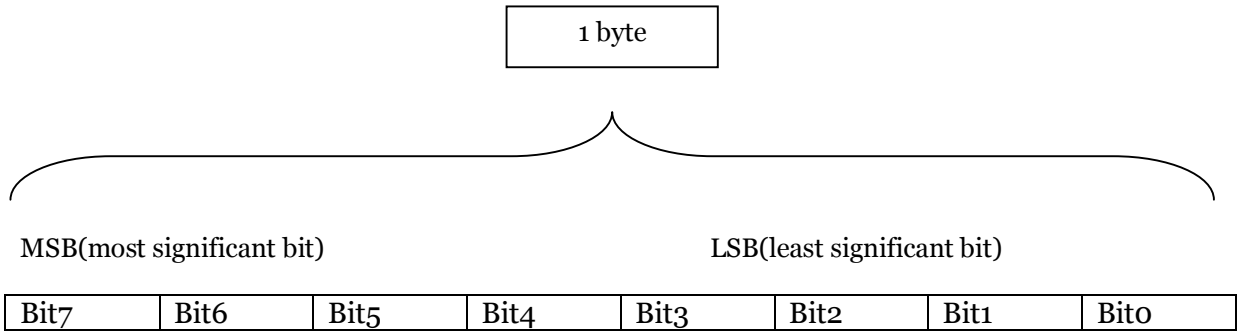
Packet unit data is defined as packet count and the kinds of data are defined as each product applying from LXSDF T2 Format. Another kind of data can be transmitted by PCE type.

Note 2

The stream data on the table shows the example of 4 channels and sample capacity transmitted per 1 packet. Also, LXSDF T2 is able to transmit channel numbers and sample capacity fluidly.

Term – digit bit, bit digit.

“digit bit” is indicated for the bit’s capacity, while “bit digit” is indicated for bit’s factor. For example, “7 bits” means “7 unit bits”, “bit 7” means most significant bit as bellowed picture. “bit 0” means most significant bit.



LXSDF T2 TX Packet Transmission Period.

The device transmits LXSDF T2 Tx Type data continuously when turning on.

When transmitting LXSDF T2 Tx packet from device, packet transmission period is set differently on each product, and transmission period is usually 128,256,521 and 1024 times per second.

TX PACKET ELEMENTS EXPLANATION.

PC (Packet Count)

Increase 1 for transmission every time and Start 0 again after the maximum value.

By using PC, it is necessarily used to identify the data transmitted to PCD every packet.

The maximum of PC value gets different value according to PCD Type value. If PCD Type =0, PC MAC is 31.

CRD (Command Response Data)

If the device receives the order of LXSDF T2 Rx format well, CRD value is reversed.

Usage – If CRD value is 1 before transmitting the order from host and the value is the same after transmitting, the order transmission is failed. If CRD value is changed, it means the device receives the order from host well.

PUD 0, PUD 1, PUD 2 (Packet Unit Data)

Allocated data is different for each product. Mainly, information data to transmit at high speed is allocated.

PCD Type (Packet Cyclic Data Type)

PACKET Count maximum value depends on this value and data transmitted to packet cyclic data depends on PCD Type value.

PCD Type value is always 0 for the first stage and though PCD mode value is changed into different value like 1,2,3.. on the situation, it is changed into 0 automatically by completing data transmission of the mode one time.

PCD Type	The maximum of PC value and Total Time	Data
0 (000)	31, (Note 1)	LXSDF T2 format exclusive data and general data (exclusive data is explained as below.)
1 (001)	Depend on each product	
2 (010)	Depend on each product	
3 (011)	Depend on each product	
4 (100)	Depend on each product	
5 (101)	Depend on each product	
6 (110)	Depend on each product	
7 (111)	Depend on each product	

Table 3. Data Batch For Each PCD Type

(Note 1) Total time.

Total time to spend is that PC value increases after transmitting packet which is PC=0 and then transmitting packet 0 again after transmitting PC MAX value.

Each product has different packet transmission period

Total Time Formula.

Transmit N Packet per second and PC MAX value is M : Total time = (1/N) x (M+1) sec

EXAMPLE: If transmitting packet 250 times per second and PC MAX value is 31, (1/250) x 32 = 0.128 sec

PCD System Designated Data of PCD Type 0.

The section from PC 0 to 23 is for transmitting specialized data and the section from PC 24 and PC 31 is for system designated data.

LIST	EXPLANATION	NOTE	PC (Packet Count).
COM PORT SEARCH INFOMTION	Information for searching device in host.	108 is fixed.	31
ComDeviceID (model number)	Discernable model number by each product. Products using LXSDF T2 Format can exist some, the model number by each product is transmitted.	Allocate value between 1 and 255	30
Firmware information	Manufacture management information	1byte.	29
Channel number	Channel number from stream area of packet	1byte.	28
Sample number	Sample number from stream area of packet	1byte..	27
Other information 1	low 3bits : Com Path	1byte.	26
Reserved	NO Batch specialized product's data.	1byte.	25
Reserved	NO Batch specialized product's data	1byte.	24

Table 4. PCD Designated Data of PCD Type 0.

Com Path of Other information 1 is allocated in low 3 bits in one byte of PCD of PC26 and is used for mark to show how to transmit data. It is possible to transmit more than 2 communication path in one device. And host received the data refers to Com path value to check the communication path.

Com Path value	Meaning	Explanation
0	WIRE UART	Data packet transmitting from device to UART
1	WIRE USB CDC	Data packet transmitting from device to USB CDC
2	WIRELESS Bluetooth SPP	Data packet transmitting from device to Bluetooth SPP

PROGRAM STRUCTURE BY HOST COMMUNICATING WITH DEVICE.

The following picture shows the device using communication format of LXSDF T2 Type and total program structure of host side. The first of all, open Com port in host. Set UART in MCU in embedded system. After that, "1. Read data from Com port" reads bytes received from com port(UART) in order. "2. Abstract LXSDF T2 Tx packet." Which abstracts sync byte (it is placed in order of 255,254) which means packet's start spot from byte row separates data up and then abstract data elements in the packet from "3. Take packet data". Data's contents transmitted into each product's element taken in the 3 phases. This information is used by getting "4. Takeinformation to provide device" phase. Data batch situation of other LXSDF T2 by each product is called Device Specialization. The next page shows the code example of phase 2 and 3 as the following picture.

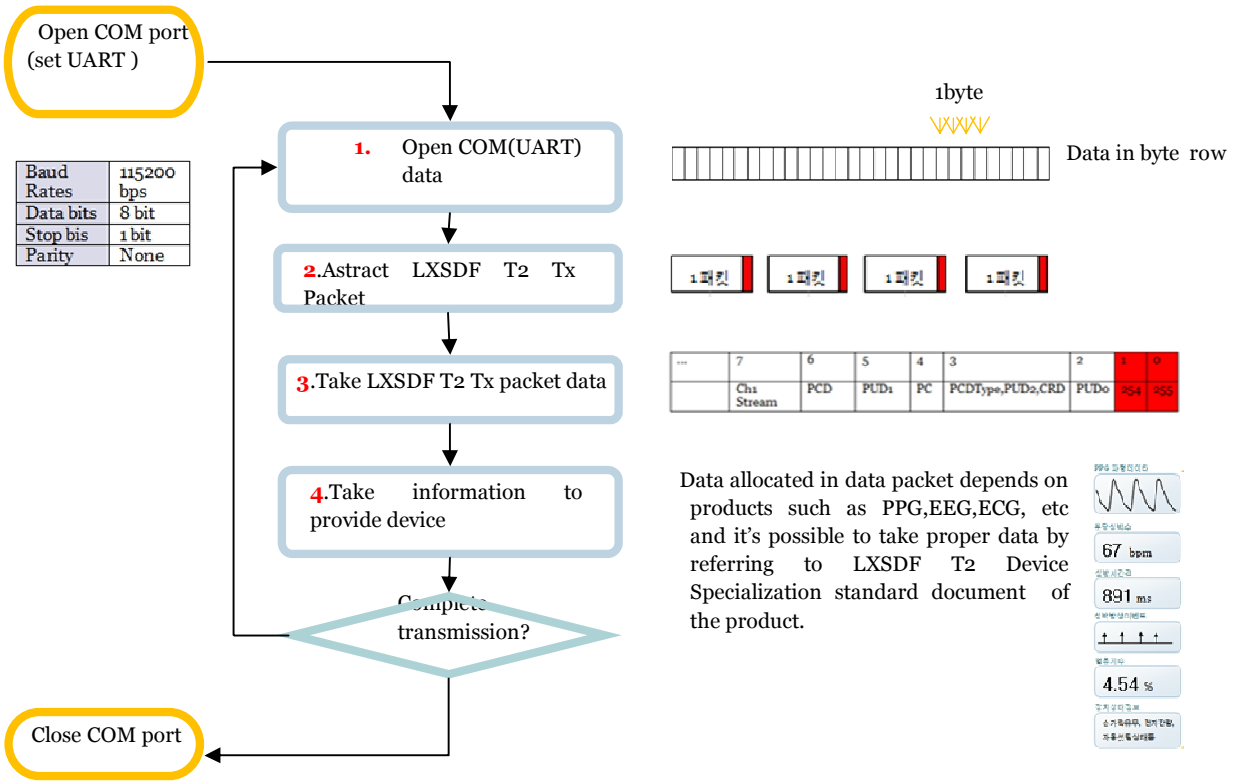


Fig 2. Total Program Structure in Host processing LXSDF T2 Tx received from device.

CODE EXAMPLE: LXSDF T2 Tx PACKET ABSTRACT AND PACKET DATA OCCUPANCY.

This code shows the example of processing code of phase 2 and 3 in the picture. (C#code). I just use simple general function so total coding method is the same regardless of the language.

```
// transmit the data received by serial with the function by 1 byte  in order .
      // processing  in the function : Find the sync spot and abstract data by each packet TX Index .
      bool Sync_After = false;
      byte Packet_TX_Index = 0;
      byte Data_Prev = 0; // PREVALUE

      byte PUD0 = 0;
      byte CRD_PUD2_PCDT = 0;
      byte PUD1 = 0;
      byte PacketCount = 0;
      byte PacketCyclicData = 0;
      byte psd_idx = 0;

      int Parsing_LXSDFT2(byte data_crnt)
      {
         int retv = 0;

         if (Data_Prev == 255 && data_crnt == 254)// Found sync spot.
         {
            Sync_After = true;
            Packet_TX_Index = 0;              // Initialize  packet  TX Index  0.
         }

         Data_Prev = data_crnt;               // receive the present value as pre value.

         if (Sync_After == true)              // only task after discovering sync.
         {
            Packet_TX_Index++;                // increase TX Index 1. The spot where is discovered 254 is 1. Whenever
receiving 1 byte as serial, it increases 1.
            if (Packet_TX_Index > 1)          // only task over TX Index  2.
            {
               if (Packet_TX_Index == 2)          // occupied TX Index2 PUD0.
                  PUD0 = data_crnt;
               else if (Packet_TX_Index == 3)     // occupied TX Index 3 CRD, PUD2, PCD Type
              CRD_PUD2_PCDT = data_crnt;
               else if (Packet_TX_Index == 4)     // occupied TX Index 4  PC.
                  PacketCount = data_crnt;
               else if (Packet_TX_Index == 5)     // occupied TX Index 5  PUD1.
                  PUD1 = data_crnt;
               else if (Packet_TX_Index == 6)     // occupied TX Index 6 PCD(Packet cyclic data) .
                  PacketCyclicData = data_crnt;
               else if (Packet_TX_Index > 6)      // Stream data(wave-pattern data) enters one each in order in more than TX
Index 7. the data procedure to receive ->  it is recorded in order of Ch 1 high byte , Ch 2 high byte low byte..
               {
                  psd_idx = (byte)(Packet_TX_Index - 7); // Packet Stream Data arrangement Index.
                  PacketStreamData[psd_idx] = data_crnt; // crnt data is occupied in order and stream data is only occupied.
                  if (Packet_TX_Index == (Ch_Num * 2 * Sample_Num + 6)) // Channel number  x 2( 2bytes occupation) x
Sample capacity + 6(Index value before the front section of wave-pattern data) is the end of one packet.
                  {
                     Sync_After = false; // Be false to search sync spot again.
                     retv = 1; // If Passing of 1 packet unit is finished, it will be returned.
                  }
               }
            } //if (Packet_TX_Index > 1)
```

```
        }
        return retv; //If 1  packet is finished, others return 0.
    }
```

TABLE5. CODE EXAMPLE: LXSDF T2 Tx PACKET ABSTRACT AND PACKET DATA OCCUPANCY

Appendix A. COM PORT AUTOMATIC SEARCH METHOD.

This explanation is for the device search in case device like PC is not fixed with some Com port number allotment. If connecting MCU and UART in Embedded System, it doesn't need automatic search.

Abstract

In case of device detecting COM port from host, there is case that user choose COM port to communicate in application program. This means bad products design regardless of user's convenience. It has to be designed that the program search COM port connected with device automatically. This function can not be solved only by software. It has to set the function to search automatically in the device.

COM port search method.

In LXSDF T2 Tx Format, PCD "Com port search information" which is PCD Type 0 and PC=31 and PCD "model number" which is PC=30 is used to find proper COM port of the product for information to use device search.

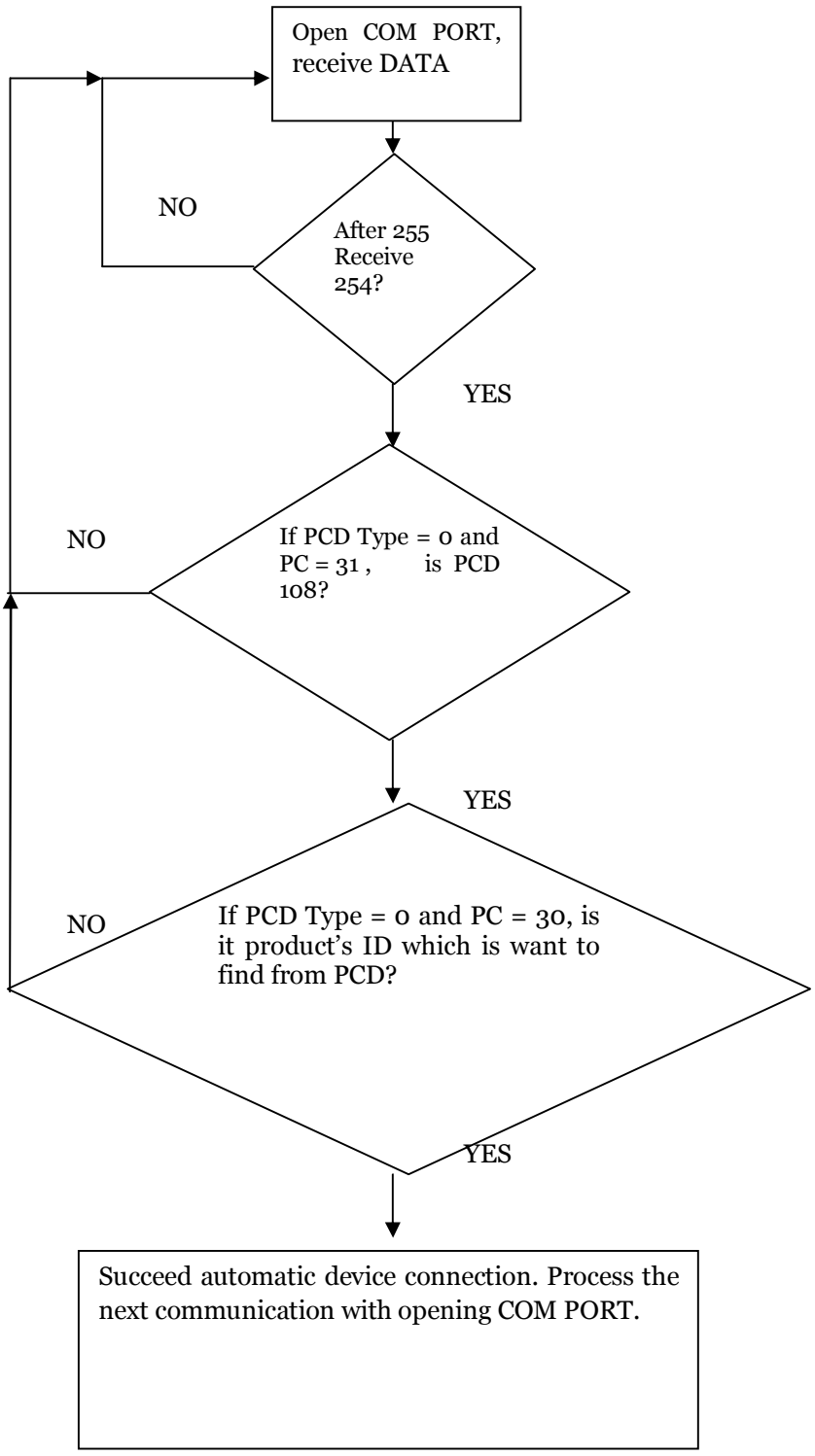
You can find the device's COM port easily if practicing the procures as the following explanation cycling all the Com port from host in order.

Open one COM port in the device temporarily and process the data received by byte unit like the following table.

Phase	EXPLANATION
Phase 1	If 254 is detected next to 255 -> It's possible device to communicate. If 254 is not detected next to 255, it's not LXSDF T2 Format. Start again opening another COM port.
Phase 2	If PC (PACKET COUNT) value becomes number 31 -> It is sure of the device transmitting data to LXSDF T2 Format. However, it could transmit the same format data like LXSDF T2 in some products coincidentally. For occupying safely, if the packet cyclic data value in PC = 31 is 108, it is sure that is 100% LXSDF T2 Format.
Phase 3	If device is communicated by LXSDF Tx format, the next phase is to search model to communicate. At this time, check packet count 30 which is product's ID to communicate.

COM PORT AUTOMETIC SEARCH FLOW EXAMPLE

‘



COM PORT AUTOMETIC SEARCH C# CODE EXAMPLE.

Autometic search method to find the devise is the same regardless of language whether it is C# or C++

```
        int bytestoread = sp.BytesToRead;   // occupied  byte number in Com port buffer. Sp is serial port object.

        // OUTPUT 1. Whether it is our device or Not? Our device must have the data in COM port..
        if (bytestoread == 0) { return; }  // If there is no data which can  be read in  COM port, this is not  LXSDF T2 format.
LXSDF T2 transmits the data every time.

        /// If there is some data to read in COM port, it reads all the data.
        byte[] rbuf = new byte[bytestoread]; // created the memory size dynamically.
        bool find_sync = false;
        sp.Read(rbuf, 0, bytestoread); //  received in rbuf tentatively..
        // OUTPUT 2. Check sync .
        for (int i = 0; i < bytestoread-1; i++) //
        {
           if (rbuf[i] == 255 && rbuf[i + 1] == 254) // Found the sync spot.
           {
              find_sync = true;
              break; // break the loop
           }
        }
        if (find_sync == false) return; // If there is no data in order of 255, 245, this is not  LXSDF T2.
        ///OUTPUT 3. Check the packet cyclic data in case of detecting some sync.  Must receive over certain time data
continuously to check it.
        byte[] cbuf = new byte[4096];
        int bytetoreadlimit =0;
        int readbytenum = 0;
        int sum_readbytenum = 0;
        bool while_continue = true;
        byte Packet_Count =0;
        byte PacketCyclicData = 0;
        bool find_108 = false;
        byte find_ComDeviceID = 0; // ComDeviceID allots more than value 1.
        byte find_NumChannel = 0;
        byte find_NumSample = 0;
        byte find_firmversion = 0;

        while (while_continue)
        {
           if(sp.BytesToRead > 4096)
              bytetoreadlimit = 4096;
           else
              bytetoreadlimit = sp.BytesToRead;

           readbytenum = sp.Read(cbuf, 0, bytetoreadlimit); // read the data and figure the byte cumulative sum.

           sum_readbytenum += readbytenum;

           for (int i = 0; i < readbytenum-3; i++)
           {
              if (cbuf[i] == 255 && cbuf[i + 1] == 254) // detected sync spot.
              {
                 Packet_Count = cbuf[i + 4];          // occupied packet count value.
                 PacketCyclicData = cbuf[i + 6];      // occupied packet cyclic data.



  if (Packet_Count == 31 &&  PacketCyclicData == 108)// If packet count is 31 and packet cyclic data is 108, it is surely
```

```
LXSDF T2 Type.
                    find_108 = true;
                else if(Packet_Count == 30)              // This spot  is for Product ID.
                    find_ComDeviceID = PacketCyclicData;
                else if (Packet_Count == 29)             // This spot is for firmware version number. It is necessary if
updating firmware by UART.
                    find_firmversion = PacketCyclicData;
                else if (Packet_Count == 28)             // Channel number transmitted into stream data.
                    find_NumChannel = PacketCyclicData;
                else if (Packet_Count == 27)
                    find_NumSample = PacketCyclicData;

                if (find_108 && find_NumSample > 0)  // This means loof break because  find_NumSample is in packet count
27 and  find_108 is in packet count 31. If both value were found , Medium value could be found.
                {
                    while_continue = false;
                    break;
                }

            }
        }
        /// Designate the maximum value to review how many data can be received in COM port.  If this value is too big, it
takes very long time to search the device. So it's good to set the small value.
        /// To search the device by LXSDF T2 type , The minimum needed data capacity  must  be  at least 32 packets. In
other words, 68bytes ( byte capacity of 1 packet) x 32 = 2176 bytes.  It's possible to exam device search information because
it has 3000 bytes enough to be 32 packets.
        ///   Formula : byte capacity of 1 packet  can find the answer as 8 bytes + 64 bytes .
        /// 8 bytes : 1 packet is 8 bytes  from Tx Index 0  to 6
        /// 64  bytes :  Stream area is  channel number * 2(bytes) * sample number, though it has different value by each
product . Because the maximum channel number allotted from LXSDF T2 is 8 and sample number is within 4, the maximum
is  64 bytes.
        /// x 32 : must receive 32 packets to communicate packet count 0 to 31.
        if (sum_readbytenum > 3000)  // Forcing  Loof  break condition.
        {
            while_continue = false;
            break;
        }
    } // while
```

TABLE 6. CODE EXAMPLE: COM PORT AUTOMETIC SEARCH.

Revision History

Release Date	Doc. ID	Description of Change
2012-12-03	LXD12 V1.1	1. ADD TABLE LABLE. ADD TABLE LIST. 2. TABLE4. ADD ComPath data list and Reserved data list in PCD designated data of PCD Type 0 Page 12 3. ADD LXSDF T2 Rx and Tx CLASSIFICATION explanation and picture Page 4
2012-06-25	LXD12 V1.0	-