LXSDF T2A

LX Serial Data Format T2A

Doc. ID. LXE10 V2

Release Date. 2018-02-07

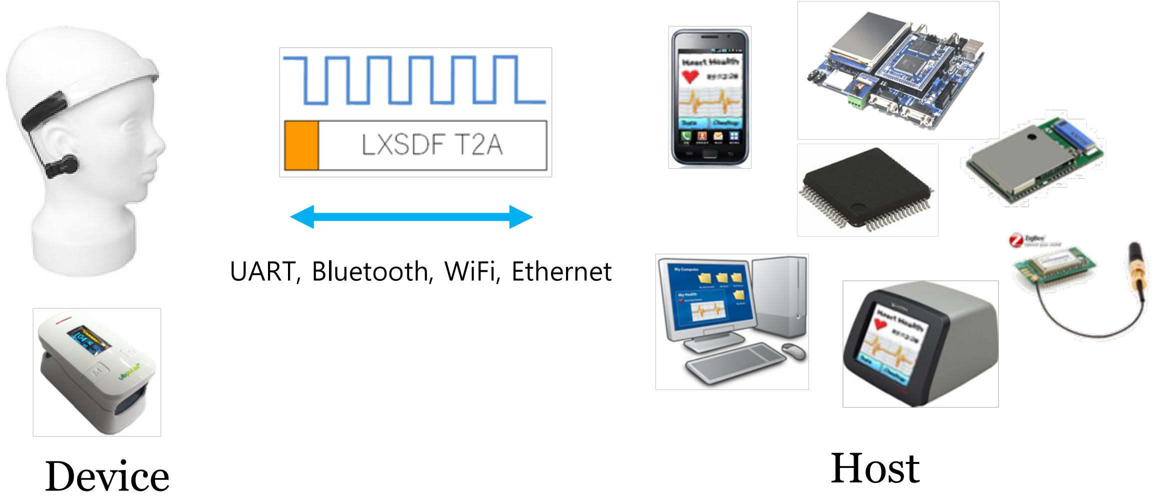
Abstract - Simple packet and protocols easy to use. UART, Bluetooth, WiFi, Ethernet, etc. Extremely simple packet and easy to use. Multi channel real time stream transmission. Support both Stream Mode and Non-Stream Mode.

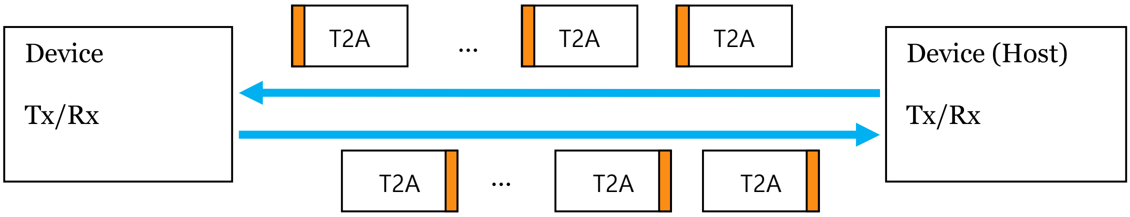
Contents

LXSDF T2A Overview

LXSDF T2A is a general purpose serial communication standard which is able to transmit real-time multi channel stream data and non-stream data. A typical example of multi channel stream is the time series data converted by the multi channel ADC(Analog to Digital Converter). LXSDF T2A packet is so simple and small packet size. LXSDF T2A can be used in any system using uart, wifi, bluetooth and ethernet etc.

UART(com port, serial port) is the most commonly used serial communication. UART can transmit 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 12 bit AD converted data, must transmit serial communication 2 bytes dividing 4 bits and 8 bits and unify them into one 12bits data in receiver. If there are various kinds of data to transmit, it is in need of packet concept. LXSDF T2A can handle several bytes as one packet, transmitting and receiving sides use the data by separating according to the standard LXSDF T2A.





T2A Packets for Stream & Non-Stream Communications

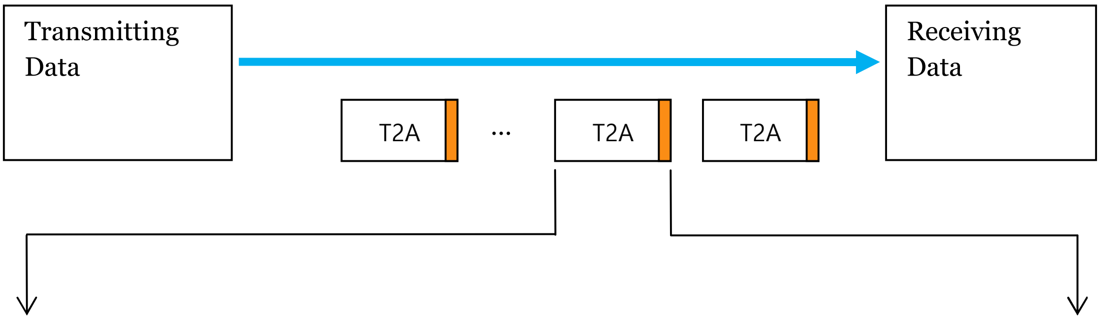
T2A Packet

Sync Bytes : the core concepts of T2A Packet

LXSDF T2A packet uses initial 2 bytes for each packet transmission as “Purpose for Synchronizing Packet”. The first byte is allocated the fixed value 255(0xFF in hex), and fixed 254 (0xFE in hex) for second byte. Namely, Sync Bytes are designed only in the spots where appear 255 and 254 sequentially in the whole packet byte arrays.

Receiving side should check each byte and extracts “Sync Bytes”, 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 orange color part in the first 1 packet is assigned “Sync. Bytes” and then a series of bytes for one packet.



…	9	8	7	6	5	4	3	2	1	0
…	PSD0	PSD1	CPP	PCD	PUD1	PC	PUD0	PPD	254	255

T2A Packet Definitions

The table shows the definitions of T2A packet. Each index is one byte and transmitted in order when transmitting serial. An available value for each packet index, the terms of the packet elements.

Index	Value	Packet Element Name for Stream Mode	Packet Element Name for Non-Stream Mode
0	255	SyncByte0 (Synchronization Byte 0)	
1	254	SyncByte1 (Synchronization Byte 1)	
2	0~254	PPD (Packet Property Data). 0~15 : Stream Mode, 16~254 : Non-Stream Mode.	
3	0~254	PUD0 (Packet Unit Data 0)	PBS (Packet Byte Size)
4	0~255	PC (Packet Count)	IID (Information ID)
5	0~253	PUD 1 (Packet Unit Data 1)	.
6	0~255	PCD (Packet Cyclic Data)	.
7	0~253	CRD (Command Response Data). bit 6 PUD2 (Packet Unit Data 2). bit 5,4,3 PCDT (Packet Cyclic Data Type). bit 2,1,0	.
8	0~253	PSD1 (Packet Stream Data High Byte)	.
9	0~255	PSD0 (Packet Stream Data Low Byte)	.
10	0~253	PSD1 (Packet Stream Data High Byte)	.
11	0~255	PSD0 (Packet Stream Data Low Byte)	.
…	…	…	…
N-1	0~253	PSD1 (Packet Stream Data High Byte)	.
N	0~255	PSD0 (Packet Stream Data Low Byte)	.

Color	Description
🟦	Common packet elements for both Stream Mode and Non-Stream Mode
🟨	Data placement for multi channel stream data. Freely expandable to any number of channels. The typical example of the multi channel stream data is the continuous output of a multi channel ADC(Analog to Digital Converter).

Difference between Stream & Non-Stream Mode

Packet Index	Stream Mode	Non-Stream Mode
2	**PPD** : available value 0 ~ 15	**PPD** : available value 16 ~ 254
3	**PUD0** : general data allocated.	**PBS** : Packet Byte Size allocated.
4	**PC** : +1 for each packet.	**IID** : Information ID allocated.
6	**PCD** : PCD data is allocated each PC.	general data allocated.
7	**PCDT** : means PCD type.	general data allocated.
8 and over	**PSD** : real time stream data allocated.	general data allocated.

Packet Byte Size

- Stream Mode : Predefined fixed value. ex. 16, 32, ..., etc depends on applications. .
- Non-Stream Mode : Determined by packet index 3(PBS) per each packet

Packet for Stream Mode

PPD (Packet Property Data)

Available 0 ~ 15 for Stream Mode.

PC (Packet Count)

+1 for every one packet transmission 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 PCDT is 0, PC maximum is 31.

CRD (Command Response Data)

If the device receives the command from the other device, 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.

PCDT (Packet Cyclic Data Type)

PC maximum depends on PCDT value and data transmitted to packet cyclic data depends on PCDT value. PCDT is always 0 for the first stage(device power ON) 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.

PCDT	PC (Packet Count) Maximum	Data
0	31	Exclusive data for LXSDF T2A and general data.
1	Depends on each products	
2	Depends on each products	
3	Depends on each products	
4	Depends on each products	
5	Depends on each products	
6	Depends on each products	
7	Depends on each products	

LXSDF T2A PCD Designated Data for PCDT 0

The section from PC 0 to 19 is for transmitting product's specialized data and the section from PC 20 to PC 31 is for system designated data. The system exclusive data is explained as below table.

PCD[PC]	Item	Description
PCD[31]	Com port search information	fixed value 109. Information for searching device using LXSDF T2A.
PCD[30]	LXDeviceID	Allocated value between 1 and 255. Unique ID for identifying the device.
PCD[29]	ComFirmInfo1	Firmware ID and version for processor 1.
PCD[28]	Number of channel	Number of channel from stream area of packet.
PCD[27]	Number of samples	Number of samples from stream area of packet.
PCD[26]	ComPath	Communication physical path.
PCD[25]	ComFirmInfo2	Firmware ID and version for processor 2.
PCD[24]	ComFirmInfo3	Firmware ID and version for processor 3.
PCD[23]	-	reserved
PCD[22]	-	reserved
PCD[21]	-	reserved
PCD[20]	-	reserved

ComPath

ComPath is used for mark to show what physical path to transmit data. It is possible to transmit more than two communication path in one device. The host received the data refers to Compath value to check the communication path.

Compath Value	Communication Path
0	UART
1	USB CDC
2	Bluetooth SPP(Serial Peripheral Profile)
3	Bluetooth Low Energy SPS

Packet for Non-Stream Mode

PPD (Packet Property Data)

Available 16 ~ 254 for Non-Stream Mode.

Predefined PPD values : 32, 34, 48, 64, 128 for designated communication types.

- 32 : Data send without request result.
- 34 : Data send with request result.
- 48 : Send result in response to PPD=34
- 64 : Request data.
- 128 : Send data in response to PPD=64.

PBS (Packet Byte Size)

Packet byte size is allocated. Packet bytes size is equal to the packet index maximum – 1.

IID (Information Identification Data)

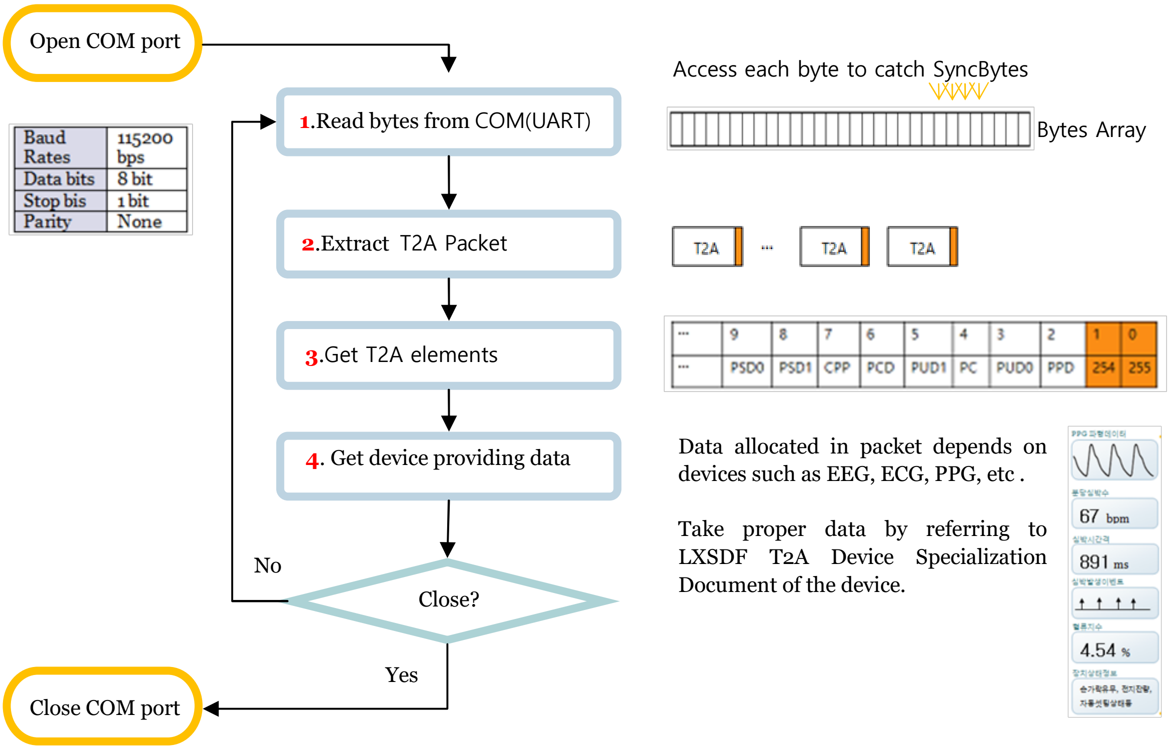
Identification number is allocated. The purpose of ID number is identifying the packet contents.

Programming Guides

The following picture shows the programming flow of host side.

First of all, open com port in host. After that,

1. Read Bytes from COM : Read bytes received from com port(UART) in order.
2. Extract T2A packet : Catch Sync Bytes (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. Get T2A element : Get all the T2A Packet elements.
4. Get device providing data : Data allocation situation depends on each device. Refer to LXSDF T2A Device Specialization Documents of the device.



Baud Rates	115200 bps
Data bits	8 bit
Stop bis	1 bit
Parity	None

…	9	8	7	6	5	4	3	2	1	0
…	PSD0	PSD1	CPP	PCD	PUD1	PC	PUD0	PPD	254	255

Data allocated in packet depends on devices such as EEG, ECG, PPG, etc .

Take proper data by referring to LXSDF T2A Device Specialization Document of the device.

Appendix A. COM PORT AUTOMATIC SEARCH METHOD.

This explanations are for the device search in case device like PC. The PC attached device's com port number assignment is not fixed. If connecting MCU and UART in Embedded System, it doesn't need automatic search.

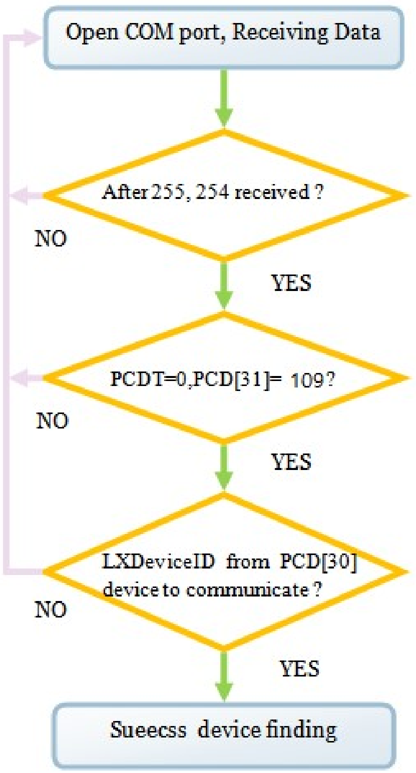
Abstract

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 packet, PCD "Com port search information" which is PCD Type 0 and PC=31 and PCD "LXDeviceID" 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 temporarily and process the data received by byte unit like the following table.

Flow Chart	Step	Description.
 Open COM port, Receiving Data After 255, 254 received ? NO YES PCDT=0,PCD[31]= 109? NO YES LXDeviceID from PCD[30] device to communicate ? NO YES Sueecss device finding	Step 1	If 254 is detected next to 255 : It's possible device to communicate. Goto step 2. If 254 is not detected next to 255 : it's not LXSDF T2A packet. Start again opening another COM port.
	Step 2	If PC (PACKET COUNT) value becomes number 31 , it is sure of the device transmitting data to LXSDF T2A Format. However, it could transmit the same format data like LXSDF T2A in some products coincidentally. For occupying safely, if the PCD[31] is 109, it is sure that is LXSDF T2A packet.
	Step 3	If device is communicated by LXSDF T2A format, the next phase is to search model to communicate. At this time, check PCD[30] which is product's LXDeviceID to communicate.

COM PORT AUTOMATIC SEARCH C# CODE EXAMPLE.

Automatic search method to find the device. The method 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_109 = 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 == 109)// If packet count is 31 and packet cyclic data is 109, it is surely
LXSDF T2 Type.
                          find_109 = 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_109 && find_NumSample > 0)  // This means loof break because  find_NumSample is in packet count
27 and  find_109 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
```

Revision History

Release Date	Doc. ID	Description of Change
2018-02-07	LXD10 V2	more clear description. Appendix A. Com port auto search added.
2017-04-01	LXD10 V1	in English
2015-08-25	LXD10 V0	First release.