

# Application Note AN-00156

## Reading the Tx ID from the MS and HS Decoders

### Introduction

The Linux MS and HS Series decoders have the ability to output a number that uniquely identifies the encoder that sent the transmission. This can be used to log access attempts or in any application where the originating transmitter needs to be known. This application note discusses how to read the ID number and make use of it. Sample software is available for download from the Linux website and is described here.

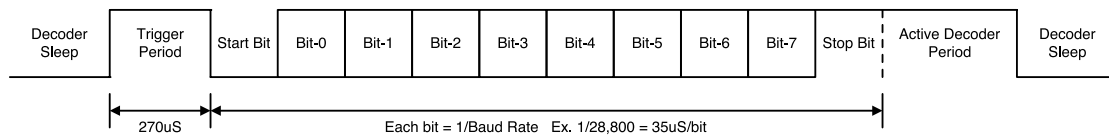
### Decoder Operation

In order for the MS and HS decoders to accept a transmission, they must first learn the encoder's Address. When the decoders receive a transmission they compare the received Address with one that is saved in memory. If a match is found, then the decoders reproduce the encoder's data line states on its data lines.

The decoders can store 40 Addresses and output the number in which the Address was learned as the Tx ID. So, for example, the first Address that is learned has an ID of '1' that is output as a binary 1 (0000 0001). The second Address is '2' and its ID is output as a binary 2 (0000 0010) and so forth.

Once a transmission is received and the decoder determines that it is valid, it outputs the number in which it was learned. This happens once after reception of the first valid packet. The ID number is output as an eight-bit byte at the baud rate that is set by the SEL\_BAUD lines.

#### MS Series Decoder TX\_ID Packet Timing



#### HS Series Decoder TX\_ID Packet Timing

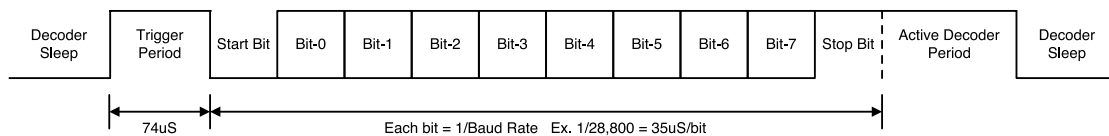


Figure 1: TX\_ID Packet Structure for the MS and HS Decoders

With this simple output, it is an easy task for a software routine to associate a number with a specific transmitter. There are three common methods of reading the ID with an RS232 serial port, USB, or an embedded microcontroller. The following section shows examples of all three.

## Examples

### Send Over RS232

Figure 2 shows the circuit schematic to send the TX ID over RS-232.

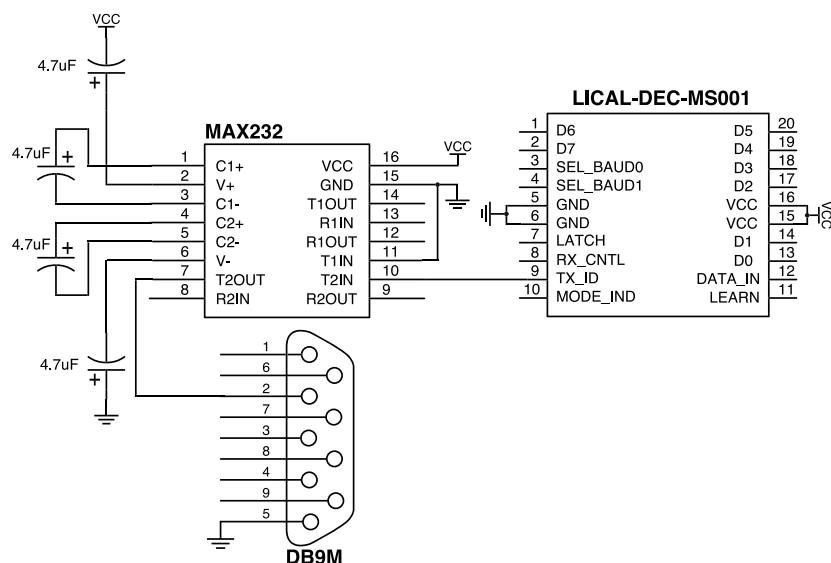


Figure 2: Schematic to Send Tx ID Over RS232

A Maxim Semiconductor MAX232 level converter chip is shown, but any RS232 level converter can be used. The TX\_ID line on the decoder is connected to one of the RS232 driver inputs on the MAX232. The output is connected to a DB9 connector, which can then be connected to a PC. Sample software to read the ID from the serial port can be downloaded from the Linx web site. This software is written in Visual Basic and is well-commented so that it can be easily included into application software or ported into other formats.

### Send Over USB

Figure 3 shows the circuit schematic to send the Tx ID over USB.

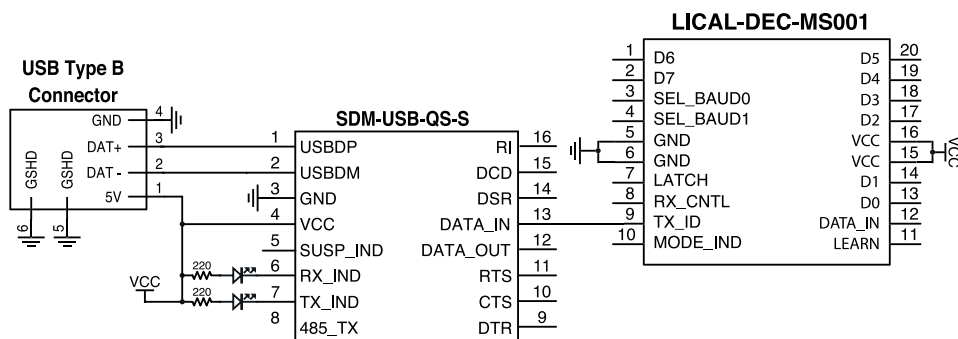


Figure 3: Schematic to Send Tx ID Over USB

The TX\_ID line of the decoder is connected to the DATA\_IN line of a Linx QS Series USB module, which is then connected to a USB type B jack. The QS module passes the data to a PC. Sample software to read the ID from the PC can be downloaded from the Linx website. This software is written in Visual Basic and is well-commented so that it can be easily included into application software or ported into other formats. It makes use of the Direct Drivers provided for the QS module. The Virtual COM Port drivers can be used with the software for use with a RS-232 serial port that was described in the previous section. This offers the designer some flexibility when working on the system design. Please see the documentation for the QS module for details on the module and drivers.

## Send to a Microcontroller

Figure 4 shows the circuit schematic to send the Tx ID to a microcontroller.

This example makes use of a PIC16F88 microcontroller from Microchip and the DMC-16202-LY LCD screen from Optrex. Both of these components are readily available from electronics distributors. The PIC16F88 was chosen because it had more than enough I/O pins and a built-in UART. The UART is specifically made to handle serial data, making it perfect to read the ID from the decoder. The software to control it is also quite simple, just telling the UART to get data at a certain baud rate.

A UART is not required, though. The controller can be programmed to sample the state of a line and watch for the individual bits at the timing determined by the baud rate. This method works and could reduce costs by allowing a less expensive controller to be used, but the software gets much more complicated and is not covered here.

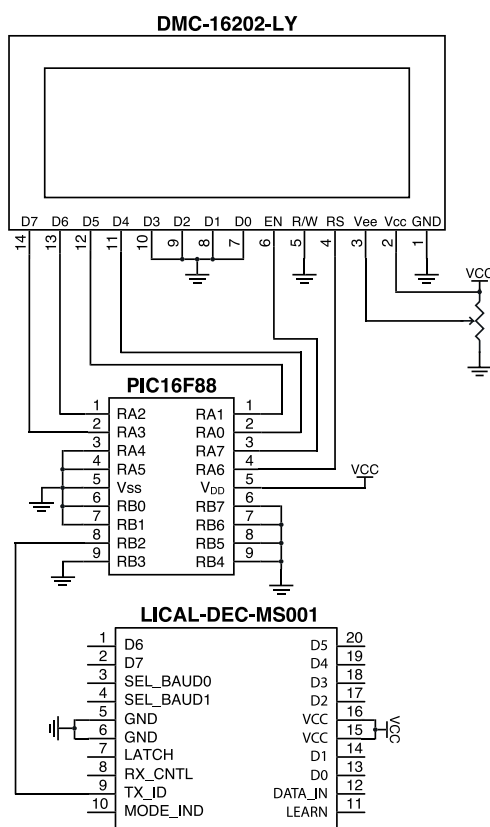


Figure 4: Schematic to Display the Tx ID to a LCD

Other microcontrollers with fewer pins, fewer features and a lower cost can be used with minor changes to the software. It is up to the designer to determine and make the changes.

The Optrex LCD screen has two lines of sixteen characters each. It takes data four bits at a time each time a control line is pulsed. Another control line tells the screen if the data is set-up information or for display. Other LCD screens can be used, but it is up to the designer to modify the software for operation with them.

The software in this example is written in C and was compiled using the PIC C compiler from CCS Info. The source code can be downloaded from the Linx Technologies web site and is well-commented, but some parts may need to be modified for different devices.

---

## TE TECHNICAL SUPPORT CENTER

USA:	+1 (800) 522-6752
Canada:	+1 (905) 475-6222
Mexico:	+52 (0) 55-1106-0800
Latin/S. America:	+54 (0) 11-4733-2200
Germany:	+49 (0) 6251-133-1999
UK:	+44 (0) 800-267666
France:	+33 (0) 1-3420-8686
Netherlands:	+31 (0) 73-6246-999
China:	+86 (0) 400-820-6015

## te.com

TE Connectivity, TE, TE connectivity (logo), Linx and Linx Technologies are trademarks owned or licensed by the TE Connectivity Ltd. family of companies. All other logos, products and/or company names referred to herein might be trademarks of their respective owners.

The information given herein, including drawings, illustrations and schematics which are intended for illustration purposes only, is believed to be reliable. However, TE Connectivity makes no warranties as to its accuracy or completeness and disclaims any liability in connection with its use. TE Connectivity's obligations shall only be as set forth in TE Connectivity's Standard Terms and Conditions of Sale for this product and in no case will TE Connectivity be liable for any incidental, indirect or consequential damages arising out of the sale, resale, use or misuse of the product. Users of TE Connectivity products should make their own evaluation to determine the suitability of each such product for the specific application.

TE Connectivity warrants to the original end user customer of its products that its products are free from defects in material and workmanship. Subject to conditions and limitations TE Connectivity will, at its option, either repair or replace any part of its products that prove defective because of improper workmanship or materials. This limited warranty is in force for the useful lifetime of the original end product into which the TE Connectivity product is installed. Useful lifetime of the original end product may vary but is not warrantied to exceed one (1) year from the original date of the end product purchase.

©2023 TE Connectivity. All Rights Reserved.

08/23    Original