

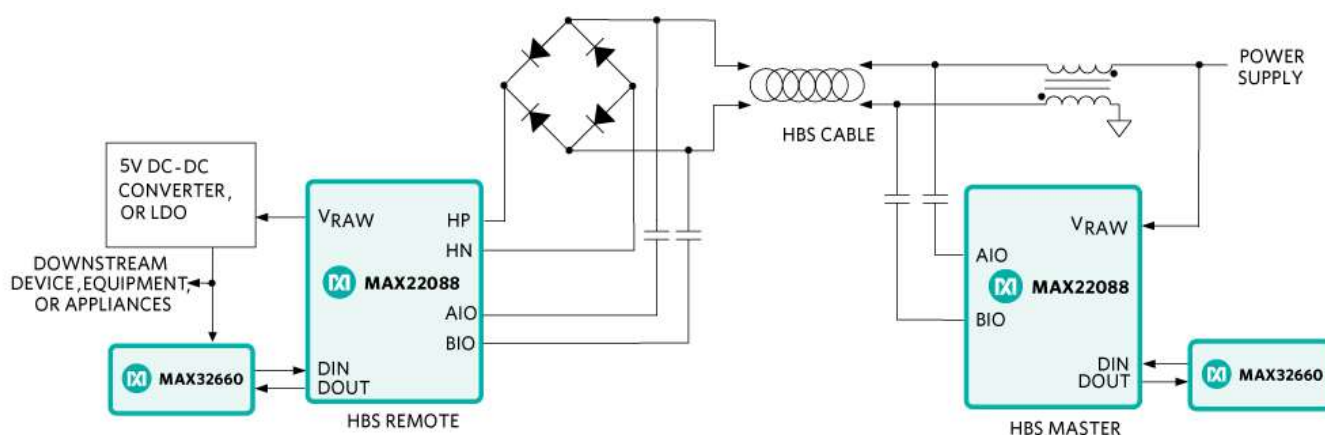
# How to Transmit UART Packets Using a Home Bus System (HBS) Compatible Transceiver

## Abstract

Maxim Integrated's first Home Bus System (HBS)-compatible transceiver, the [MAX22088](#), has improved performance for harsh industrial environments. Designed for Home Bus systems, the MAX22088 is not limited to those networks. This application note demonstrates an easy solution to encode and transmit the UART data packets using the MAX22088 HBS-compatible transceiver.

## Introduction

Maxim Integrated recently introduced the MAX22088, its first Home Bus System (HBS) compatible transceiver. The MAX22088 complies with the HBS standard but can also be used in other types of communication systems. This application note demonstrates how to generate and transmit UART packets using the MAX22088 HBS-compatible transceiver and the [MAX32660](#) microcontroller. The MAX32660 microcontroller is used to generate UART data packets for the master transceiver and another MAX32660 is used to decode the data received at the remote transceiver (Figure 1). The method was verified using two MAX22088 EV Kits; test data, and code examples are presented and discussed.



**Figure 1. Simplified Block Diagram**

## The Home Bus System (HBS) Standard

In today's home automation (HA) systems, information exchange among multiple appliances, equipment, and control devices requires a standardized communication protocol for effective and robust data transmission. The HBS standard simplifies communication among these devices by connecting them to a single bus, meanwhile defining a transmission standard that delivers power and exchanges data at the same time.

The MAX22088 HBS-compatible transceiver supports transmitting data and power on a single pair of wires for HBS applications such as HVAC, and remote monitoring and sensing. A standard HBS wiring scheme does not allow for a master clock signal to synchronize the input and output data bits. Instead, operating like the standard serial UART communication, data transitions are detected on the bus and are decoded by the receiving circuit.

## Transmitting UART Packets Using the MAX22088 HBS Transceiver

The UART protocol is an asynchronous serial data communication standard. Instead of a master clock synchronizing the output bits, UARTs use start and stop bits in the packet to detect data being transmitted. UART communication is a common integrated function of most microcontrollers, including the MAX32660.

Standard HBS cables carry both power and data from the controller to downstream remote devices/nodes. HBS data transmission uses Alternative Mark Inversion (AMI) and negative logic encoding with a 50% duty cycle in which bipolar pulses represent a logic "0." A logic "1" occurs when the transceiver outputs are in a high-impedance, or inactive state. It is a requirement for Home Bus communication that there can never be two simultaneous logic "0" conditions on the bus. This is to avoid saturation and power problems on the line as data is transmitted. The firmware, or communication software, is primarily responsible for ensuring that this requirement is met.

For this reason, the UART data packets need to ensure every logic "0" must be followed by a logic "1" in a normal Home Bus system. One easy encoding solution is to stuff every other bit with a logic "1." This method doubles the packet length and halves the transmission frequency of the UART communication. Note that HBS-compliant communication would require a parity bit following a character frame, however this method does not include one.

The MAX32660 microcontroller UART output includes 8 bits of data: a start bit, and a stop bit, but does not include a parity bit. The 8-bit data expands to a total of two packets and are transferred over the Home Bus (Figure 2).

The code presented below is used to encode the UART packet. It splits and assigns the 8-bit data to even bits of the packet and assigns a logical "1" to odd bits of the packet.

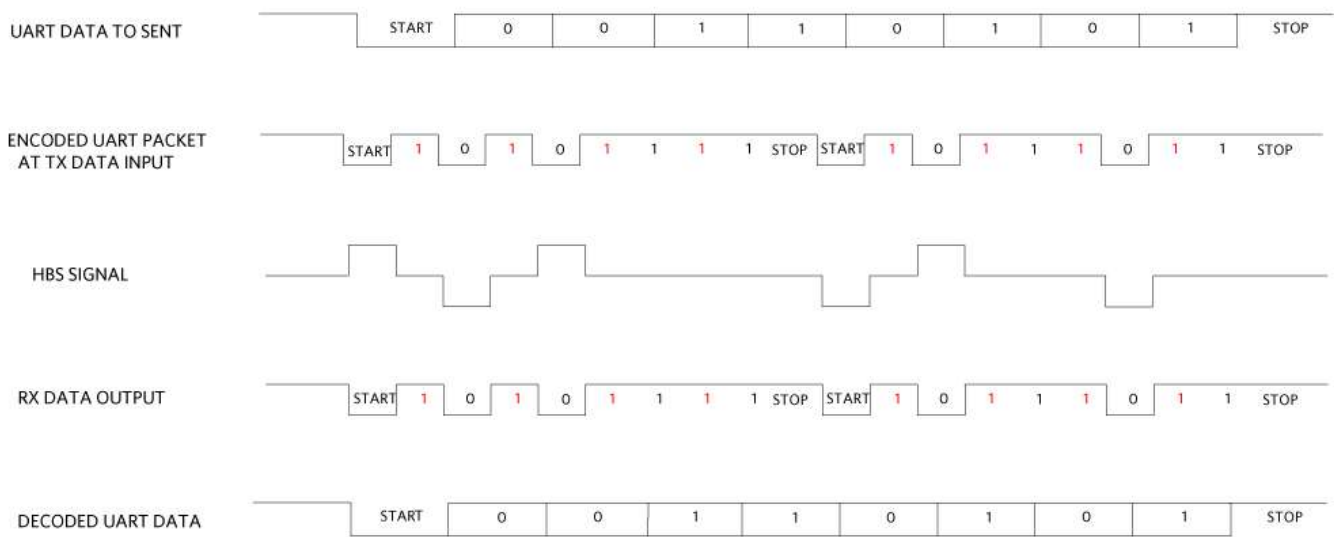
```
for (i = 0; i < TX_Length; i++)
{
    txdata[(i*2) + 0] = (( tx_text[i] & 0x08) << 3 ) +
                        (( tx_text[i] & 0x04) << 2 ) +
                        (( tx_text[i] & 0x02) << 1 ) +
                        (( tx_text[i] & 0x01)) + 0xaa;

    txdata[(i*2) + 1] = (( tx_text[i] & 0x80) >> 1 ) +
                        (( tx_text[i] & 0x40) >> 2 ) +
                        (( tx_text[i] & 0x20) >> 3 ) +
                        (( tx_text[i] & 0x10) >> 4 ) + 0xaa;
}
```

For example, if the data (tx\_text[i]) to be sent is 0x35 (0b'00110101), the encoded packet (txdata[i]) will be 0b'10101111, 0b'10111011.

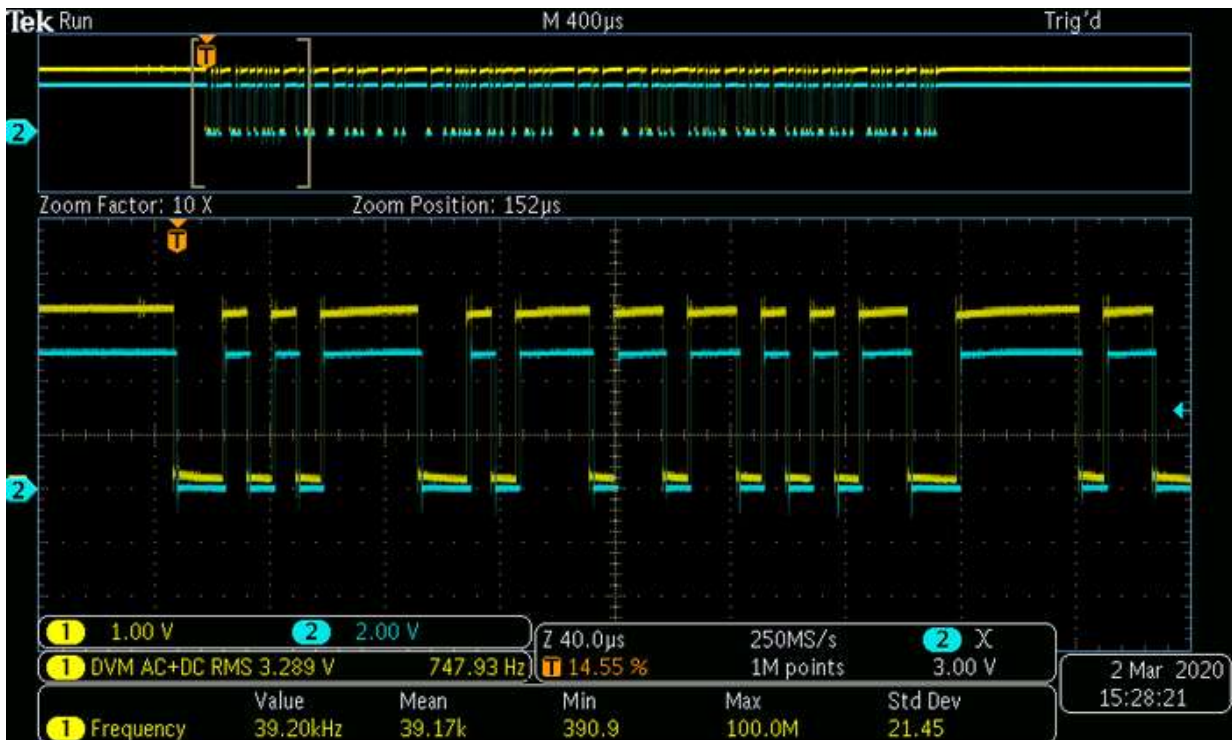
SHARES





**Figure 2. Encoded UART Data Packets Sent Over Home Bus**

Figure 3 shows a successful transmission of the UART data 0x35 using the Home Bus with encoding.



**Figure 3. Transmission of 0x35 (0b'00110101) in Home Bus with Encoding**

The UART data is encoded such that every "0" is followed by a "1" and data is properly transmitted and received from the bus. The data input to the HBS transmitter and data output from the HBS receiver match; and the MAX32660 at the receiver side decodes the output bits with no errors.

## References

Application Note 7224: Introduction to Home Bus, Shasta Thomas.

SHARES



**SOCIAL**



**QUICK LINKS**

- |                        |                                  |
|------------------------|----------------------------------|
| <b>About ADI</b>       | <b>Investor Relations</b>        |
| <b>ADI Signals+</b>    | <b>News Room</b>                 |
| <b>Analog Dialogue</b> | <b>Quality &amp; Reliability</b> |
| <b>Careers</b>         | <b>Sales &amp; Distribution</b>  |
| <b>Contact us</b>      | <b>Incubators</b>                |

**LANGUAGES**

- English**
- 简体中文**
- 日本語**

**NEWSLETTER**

Interested in the latest news and articles about ADI products, design tools, training and events? Subscribe today!

[Sign Up](#)