May 16, 2006

# Serial Instrument Bus Protocol

R.G. Lueck

**Product reference:** P038, P039

## Introduction

The purpose of this document is to explain the signalling protocol used with the 16-bit bi-directional Serial Instrument Bus of the ODAS system. This bus is used with all small form-factor instruments manufactured by Rockland Scientific International, Inc.

## Overview

The Serial Instrument Bus is used to communicate addresses (which are requests for data) and data between devices that can produce data (such as an analog-to-digital converter) and a bus controller. In a telemetering instrument (which has a communication line to the ship), the bus is controlled by a remote serial transceiver (RSTRANS). The RSTRANS also handles all communication between an instrument and a master computer located on a ship using a 2-wire communication link. This link can be several kilometers long. Further information of the remote transceiver can be found in Application Note AN 008 “ODAS Serial Lines”.

Internally recording instruments are designed to be used where a communication link between the instrument and the ship is impossible. The ship-board master computer and the RSTRANS are replaced by a recorder-controller board that is located inside of the instrument. This recorder-controller manages the serial instrument bus.

A telemetering instrument is a real-time instrument that is under the complete control of the ship-board master computer. The master computer transmits 8-bit addresses and expect to get a reply of one 16-bit data word after each address. It is imperative that every address reaches one (and only one) device that is capable of responding. By assigning the signal produced by a sensor to an address, it is possible to sample the signals from a remote instrument. If a sensor produces more than one signal, then each of its signals must be assigned to a unique address. For example, the electronics for a thermistor produce both a signal that is (roughly) liner with a respect to temperature and also a signal that is related to the temperature plus its time derivative. These two signal go to separate channels on the analog-to-digital converter. Each can be addressed to obtain its data. Data can also be produced, for example, by a frequency counter (used with sea-Bird sensors) as well as a compass.

The above described procedure also occurs in an internally recording instrument. The recorder-controller generates the addresses and expect to receive a data reply to every address. The data are buffered and then stored in a Flash memory chip.

**Details**

The serial instrument bus (S_I_Bus) consists of 14 lines (Figure 1**Error! Reference source not found.**). Two (2) of these lines are used to provide 3.3V power to the peripherals and three (3) more are used for power and signal return (DGND). The peripherals, such as the analog-to-digital converter, use these lines to energize their digital electronics. The remaining lines are used to transmit the addresses and data and to control their flow.

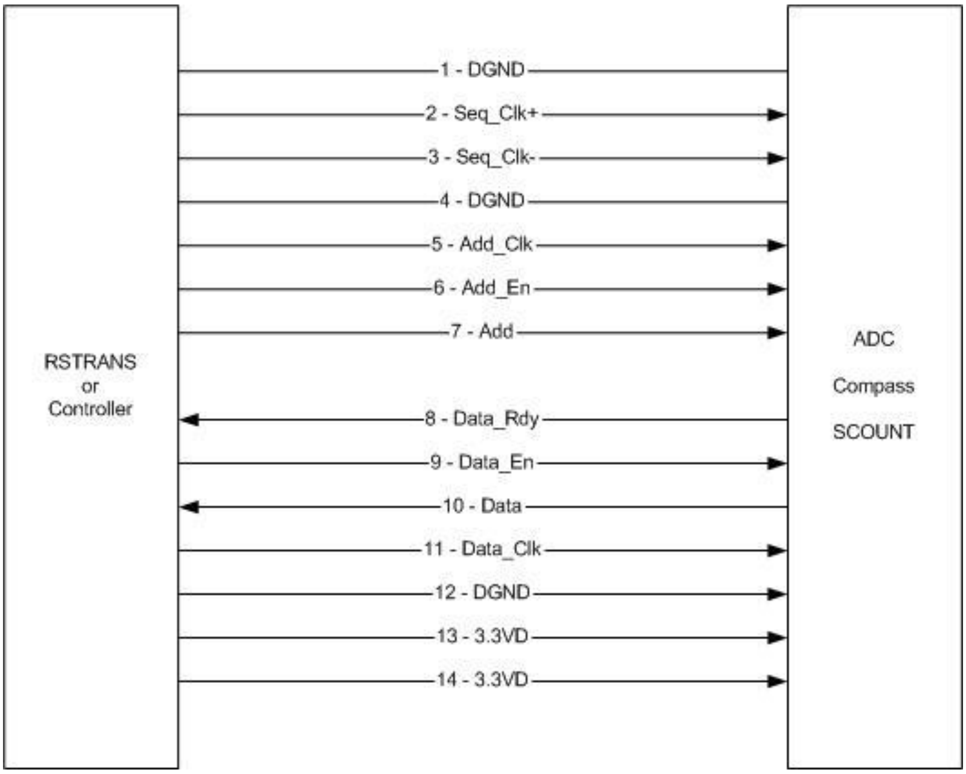

**Figure 2.** Detail of the ODAS Serial Instrument Bus. The RSTRANS (Remote Serial TRANSceiver on a telemetering instrument) or the Controller-Recorder (on an internally recording instrument) controls the bus. Power to operate digital devices is supplied by lines 13-14). A sequence clock (Seq_Clk, 1.6 MHz) is provided for internal use by the digital devices. An Address (Add), which constitutes a request for a data word, is transmitted to the digital devices using lines 5-7. The digital devices indicates that its data word is ready by raising Data_Rdy (line 8). The RSTRANS initiates the transfer of this data word using lines 9 - 11). The arrows indicate the direction of the flow of energy.

The timing for the transfer of addresses from the RSTRANS (or Recorder-controller) is shown in Figure 3. The most significant bit is transferred first. All three lines (Add_En, Data_Clk, and Add) are driven by the RSTRANS (or recorder-controller).

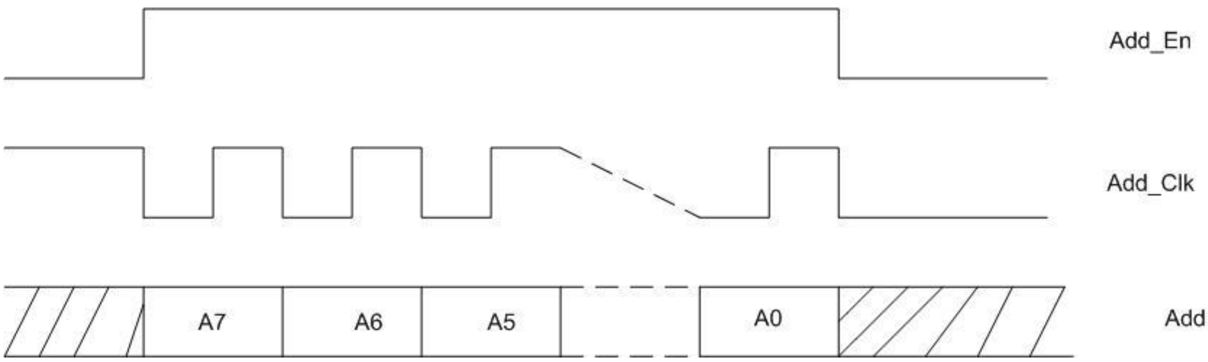

**Figure 3.** Timing diagram of an address transmitted to a peripheral digital devices. Add_En frames the transaction. The address bits (A7 – A0) are read by the peripheral device on the rising edge of the transfer clock (Add_Clk) and are updated on the negative edge. The hash marks indicate indeterminate values which can be ignored.

The timing for the data reply from the peripheral device is shown in Figure 4. This transfer involves 4 lines. The peripheral device indicates that its data word is ready by raising the Data_Rdy line. Simultaneously, the most significant data bit is placed on the Data line. When RSTRANS (or the recorder-controller) is ready to accept the data, it raises the Data_En line. RSTRANS reads the data bits on the rising edge of the Data_Clk and the peripheral must update the bits on the falling edge. Shortly after the last rising edge, which is used to read the least significant bit, the Data_En line goes low to end the transfer.

In a data transfer, the Data_Clk and the Data_En lines are driven by the RSTRANS. The Data and Data_RDY lines are driven by the peripheral device. Because there can be more than one peripheral device, these lines must be shared. To enable line sharing, the peripheral devices must hold there outputs in a high-impedance state whenever they are not involved in a transfer. The Data_Rdy line is pulled low by a resistor on the RSTRANS so that the peripheral can force the line from a low to a high state. The peripheral device should energizes the Data_Rdy line into a low state before raining this line, and it should drive it back to a low state before returning it to the high-impedance state. This will produce clean and sharp edges on the Data_Rdy line. The Data line is not terminated and, therefore, floats when there is no data transfer. This is indicated by a line at intermediate level in Figure 4.

The RSTRANS uses the rising edge of Data_Rdy to prepare itself for reading the data. The peripheral device can return the Data_RDY line to high-impedance state (which brings it low because of the pull-down resistor) at any time but not later than when Data_En goes low.

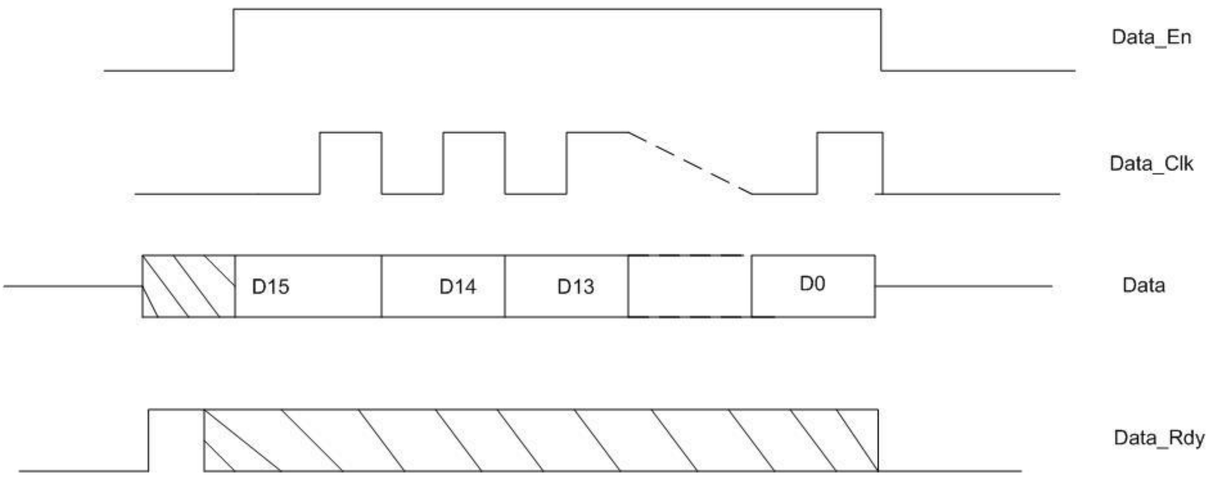

**Figure 4.** Timing diagram of the data transmitted from a peripheral digital device. The peripheral device indicates that its data is ready by raising the Data_Rdy flag. This line is tied low on the RSTRANS or Controller-recorder so that it can be shared by all peripheral devices. The peripheral device presents the data but leaves the Data line in high-impedance state (indicated by the hash lines) until the SRTRANS raises the Data_En line. The first bit is the most significant Data bit. When the RSTRANS or Controller-recorder is ready to accept the data bits, it raises the data enable line (Data_En) and reads the data bits on the rising edge of the transfer clock (Xfer_Clk). The peripheral device updates the data bits on the falling edges of the transfer clock. The peripheral device must set its data line into tri-state when (or before) the Data_En line goes low (this is indicated by the has lines).

### Sequence Clock

A continuous clock signal of 1.67 MHz (24/16 MHz) is also distributed on the Seq_Clk+/- lines. These two lines are 180 degrees out of phase to “balance” the serial bus. Only the Seq_Clk+ line is actually sensed by the logic circuits on the peripheral boards. This signal is used to sequence the activity on the peripheral boards.

The sequence clock is derived from the same master 24 MHz oscillator that is used to generate the baud rate clock which is used for the asynchronous transmission of Manchester encoded signals over the communication link to the ship-board computer. However, because the Manchester communication is asynchronous, the sequence clock is not phase-locked to the baud rate clock. See Application Note AN008 for further details on the transmission of information over the communication link between the instrument and the ship-board computer.

### Further Notes

The reading and updating of the data bits is controlled by the RSTRANS (or Recorder-controller) rather than by the peripherals for a particular reason. If RSTRANS controls the timing, then the transfers can be synchronized with the transmissions over the communication link to the ship-board computer. The transmission (to and from the ship-board computer) uses differential Manchester II encoding, sometimes known as return-to-zero encoding. This sort of encoding is required for long transmission lines and it is asynchronous. It has a large logic overhead and is power consumptive. It is not suitable for short (local) communication. The serial instrument bus uses a synchronous, no-return-to-zero, and single-ended format. The Add_Clk and Data_Clk clocks are synchronized to the Manchester II stream so that the information flows through the RSTRANS with at most a delay of ½ bit. The serial instrument bus is, thus, about as quick as the bi-directional 16-bit parallel bus used on large form-factor instruments (Application Note AN 009).

End-of-Document