Asynchronous RS-232 Communication for AEAS-7000

Application Note 5103

Introduction

The AEAS-7000 is a 16-bit gray code Absolute Encoder backbone with SSI (Synchronous-Serial-Interface) communication interface. The SSI interface is a synchronous serial communication where the master (host) and slave (AEAS-7000) devices are synchronous to each other and the data is output 1-bit per clock cycle.

This application note discusses how the additional hardware can be added to make the AEAS-7000 communicate in Asynchronous RS-232, which is available in most personal computers. The data is available to the host on demand only. RS-232 is simple, universal, well understood, and supported in most personal computers. The communication length line is restricted to less than 15 meters (50 feet) but additional hardware can be added to increase the distance. This will be discussed in further detail under the design considerations section.

Communication Protoco Synchronous Serial Interface (SSI)

SSI is a serial communication format where the master and slave devices are synchronous to each other. The master device will provide clock signal to the slave device and 1-bit of data will either output from the slave or input to the slave per clock cycle. For example, to transfer 16-bit of data, 16-clock cycle is required to move the data. The data is latched on either the falling edge or rising edge of the clock signal based on the device setup. The communication speed is based on the maximum bandwidth of the slave device. The SSI communication speed of the AEAS-7000 is up to 16 MHz, meaning that it only takes 62.5 ns to transfer 1 bit of data. But the communication speed varies depending on the master device with which it communicates. Figure 1 illustrates the SSI communication state diagram for AEAS-7000 absolute encoder.

Based on the SSI communication state diagram shown in Figure 1, the data is available at the AEAS-7000 output at the falling edge of the clock (SCL) signal. The clock high state is an idle state and the clock low state is an active state. The chip select (NSL) line is required to pull down in order for the communication to be active otherwise no activities are observed even though the clock (SCL) signal exits.

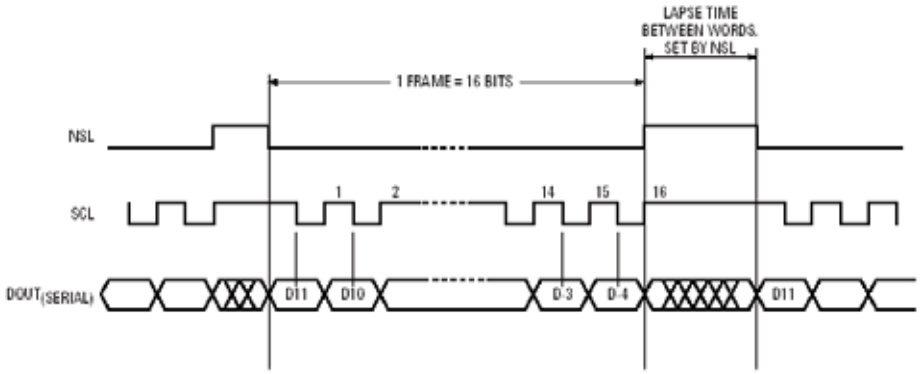


Figure 1. AEAS-7000 SSI Communication State Diagram.

Asynchronous RS-232 Interface

RS-232 is a simple, universal, very well understood interface, and is supported by most personal computers. RS-232 data is bi-polar where the voltage +3 volt to +12 volt is considered "ON" state and the voltage –3 volt to –12 volt is considered "OFF" state. Most modern computers choose to ignore the negative voltage level and consider the zero voltage as "OFF" state and same goes to the "ON" state with lesser voltage potential. With this method, the circuits powered by a 5 volt power supply are capable of driving RS-232 circuits directly but the overall range that the RS-232 signal transmitted/received may be severely reduced. Due to this, the original RS-232 voltage level is used in this design to maintain the RS-232 communication capability.

The output signal level usually swings between –12 volt and +12 volt depending on the RS-232 transceiver used in the design. The Sipex SP233 RS-232 transceiver used in this design swings between –10 volt and +10 volt. The range between –3 volt and +3 volt is considered "Dead Area" designed to absorb line noise. Figure 2 illustrates the RS-232 voltage level.

The data is received and transmitted on pins 2 and 3 respectively. One frame of data consists of one Start-Bit and one Stop-Bit. The 8-bit data is wrapped around the Start and Stop bits. If either Start-Bit or Stop-Bit is not detected during the RS-232 transmission, a framing error occurs. In this design, hardware flow control is used via RTS/CTS port. The RTS (Request-to-Send) and CTS (Clear-to-Send) are on pins 7 and 8 respectively. In the asynchronous communication the RTS and CTS signals are constantly on throughout the communication. The RTS is commonly used for turning on/off the slave device. It is important in a multi-slave system that only one slave communicates with the host at a time. When the host wants to communicate with the slave device, it raises the RTS line from –10 volt to +10 volt. Once the slave device detects the RTS signal, it raises the CTS line as acknowledgement and communication starts. When the communication is completed, it drops both the RTS line and the CTS line. Figure 3 illustrates the RS-232 common pin out.

There are several shortcomings of the RS-232 communications. First, the device situated on different electrical bus tends to have "Uncommon Ground" event. Second, in a single-ended communication, it is impossible to screen effectively for noise. By screening the entire cable one can reduce the influence of outside noise, but internally generated noise remains a problem. As the baud rate and line length increase, the effect of capacitance between the cables

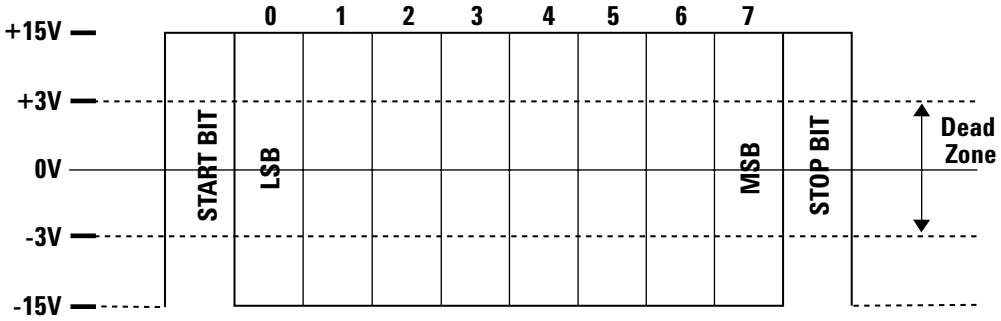
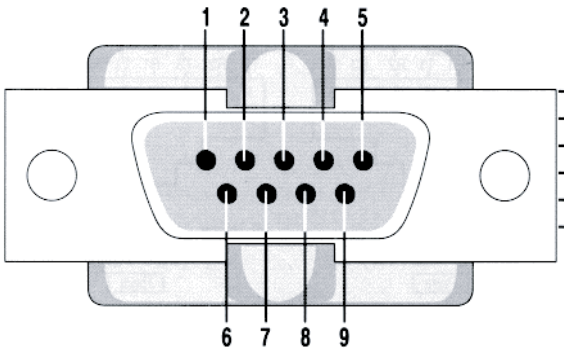


Figure 2. RS-232 Voltage Level.



Pin	Signal	Pin	Signal
1	Data Carrier Detect	6	Data Set Ready
2	Received Data	7	Request to Send
3	Transmitted Data	8	Clear to Send
4	Data Terminal Ready	9	Ring Indicator
5	Signal Ground		

Figure 3. RS-232 Pin Out.

introduces serious crosstalk until a point is reached where the data itself is unreadable.

Use of the balance line interface, RS-422/485, can eliminate these shortcomings. As a differential voltage, in principle the interface is unaffected by differences in ground voltage between sender and receiver. Furthermore, if both lines are close together, they will be affected almost identically by external electromagnetic noise. If the lines are also twisted together, then neither line is permanently closer to a noise source than the other. Hence the well known "twisted pair" is extremely effective in eliminating noise from the signal. With using the balance line interface, RS-422/485, and a high-grade cable (individually shield low capacitance pairs), the distance can be extended up to ~1200 meter (~4000 feet).

System Design Considerations

The Microchip PIC18F252 is used as the communication medium between the AEAS-7000 Absolute Encoder and the RS-232 port. The data demand from the host via the RS-232 will be acquired from the AEAS-7000 and sent to the host via the RS-232 port. The baud rate for the system is set at 19200 bauds per second, which is the supported baud speed throughout all known personal computers' RS-232 ports. Each of the AEAS-7000 Absolute Encoders with the added RS-232 hardware is address programmable and up to 15 unique addresses can be assigned. For more details about configuring the device addresses, please refer to the Application Usage section. Figure 3a illustrates the overall system layout of AEAS-7000 with RS-232 hardware features.

Based on Figure 3a, each of the AEAS-7000 Absolute Encoders can be assigned with unique addresses (00H-0EH).

The system contains two boards, a RS-232 Adapter board and a RS-232/SSI Converter board. The RS-232 Adapter board is basically a level translator to CMOS compatible level. The RS-232/SSI Converter board resides at each of the AEAS-7000s that contain an internal EEPROM to store the device addresses. Once configured the address will remain even though the device is powered down. The default device address is FFH.

The RS-232 Adapter board and the RS-232/SSI Converter board contain an internal voltage regulator to step down the 12V power supply voltage to 5V, 100 mA.

Application Usage

Programming Device Addresses

To program the device addresses, Opcode 34H is sent followed by the Device Address (00H – 0EH). The default device address is FFH. Figure 5 illustrates the device address programming state diagram.

Reading Device Addresses

To read the device addresses, Opcode 30H is sent and the Device Address is fetched and transmitted to the host. Figure 6 illustrates the device address reading state diagram.

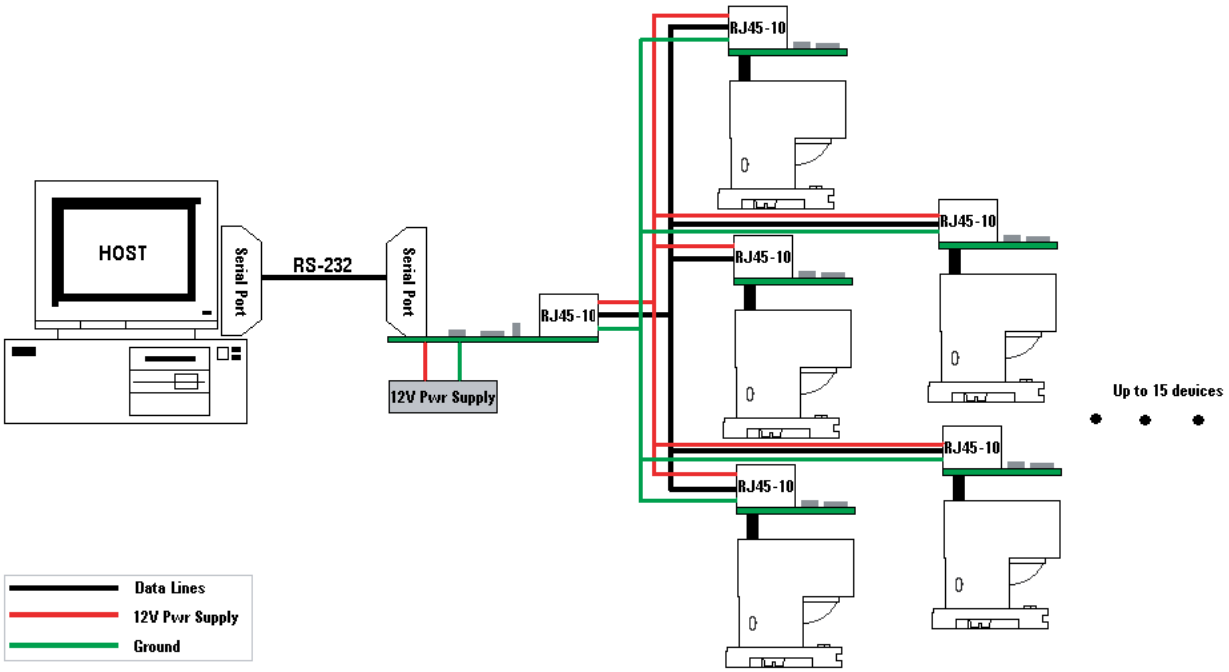
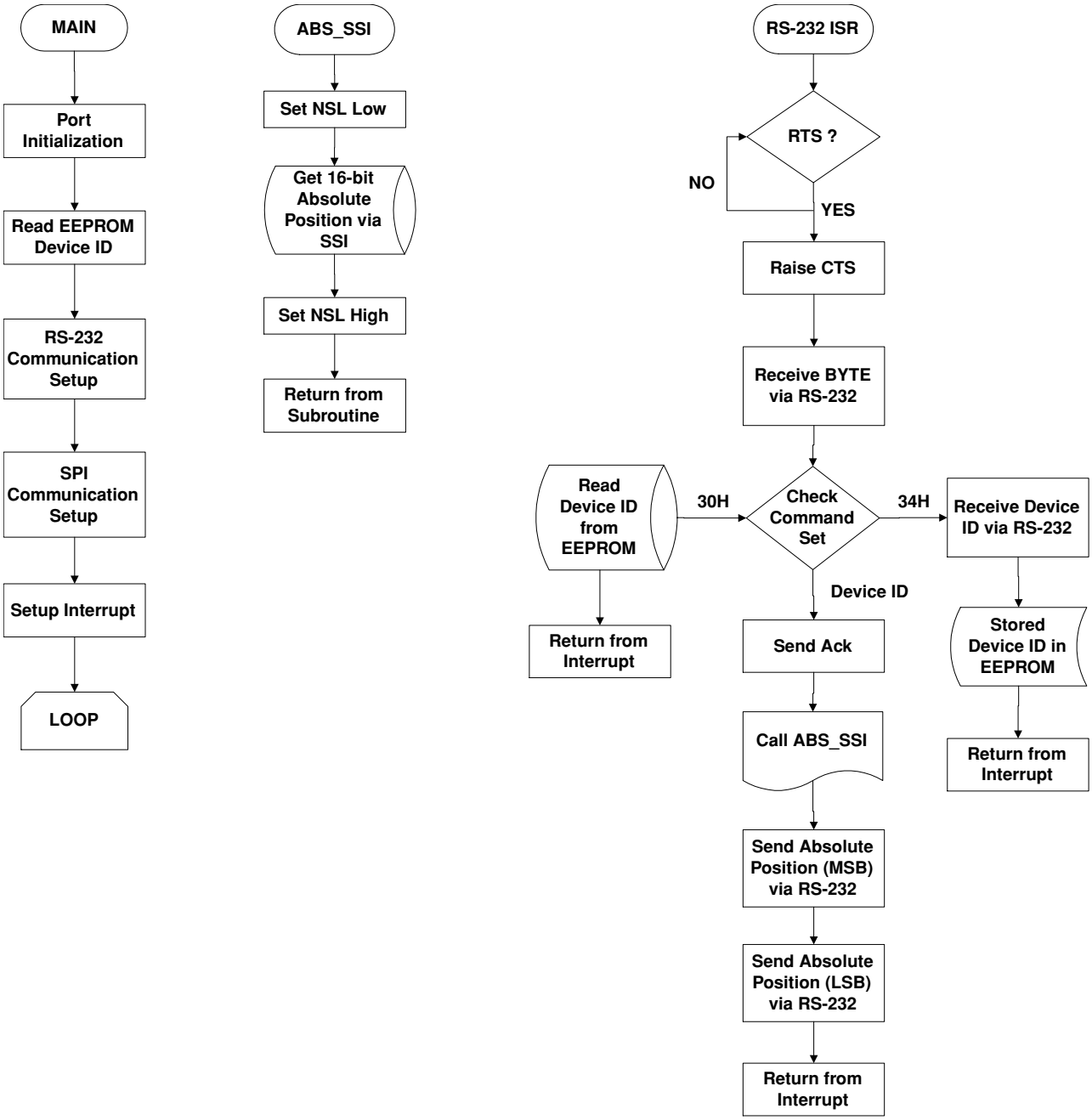


Figure 3a. Overall System Layout.

Program Flow Charts



Acquiring AEAS-7000 Absolute Position

To acquire the AEAS-7000 Absolute Position, the address of which the device absolute position to acquire is sent. If the address exists in the system, the device that is having the address returns the device address as an acknowledgement to the host. Once the communication is established between the host and the slave device, the absolute position is sent to the host with the MSB first followed by the LSB. Figure 7 illustrates the acquiring absolute position state diagram.

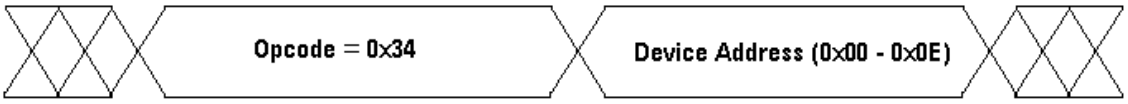


Figure 5. Device Address Programming State Diagram.

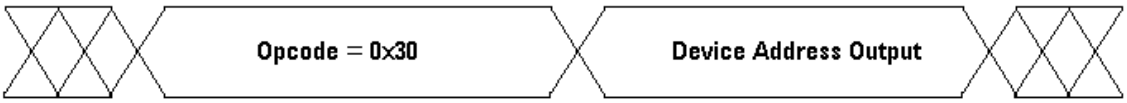


Figure 6. Device Address Reading State Diagram.

Figure 7. Acquiring Absolute Position State Diagram.

For product information and a complete list of distributors, please go to our web site: **www.avagotech.com**

5989-2277EN - October 18, 2007