ABSTRACT

Limiting this analysis to the ATMega16/32 (tentative microcontroller for the OBPP) focuses the research onto three distinct protocols: USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter), SPI (Serial Peripheral Interface), and Two-Wire Serial Interface (TWI or I2C). Each of these is an established protocol used in many microcontroller environments, but I2C (and to a lesser extent, SPI) warrants the most research and development.

TECHNICAL FINDINGS

The criterion for evaluating a communication protocol has several aspects. Firstly, the number of connections that must be made for successful communication is important considering the limitations imposed by our mechanical design. Ease of use and ease of electrical isolation are important factors as well. Finally, a discussion of actual protocol (whether it is bidirectional, addressed, framed, synchronous or asynchronous) is an important aspect.

First, we examine the SPI protocol. This is a three or four wire interface (clock, data, and slave select) that has a master/slave configuration, where the master will use the slave select line to indicate which slave it is directing transmission at, and the slaves only respond to queries, rather than full duplex (unless the four wire configuration is used). Even with the additional wire, the slave does not initiate communication. Also, because addressing is done with a slave select line, our master controller would either need 16 independent slave select lines or use an addressing scheme in a higher software layer.

The USART interface is more flexible, with a minimum of 3 connections for full duplex synchronous transmission. Still, all of the data is on serial busses, so true full duplex communication between the master and slaves is unfeasible.

There is some nomenclature confusion with the final protocol discussed here. According to the ATMega16 documentation, it is called Two-wire Serial Interface. This protocol is much more commonly referred to as I2C, or Inter-Integrated Circuit protocol. The name change might be an attempt to avoid a copyrighted name, but whatever the reason, the underlying idea is the same. This method of communication requires two or three busses (clock, data, ground) which are pulled up to power supply voltage by resistors. Open drain/collector pins can selectively pull the line down to ground to create a signal. The actual data line is completely bidirectional, which poses unique problems when trying to electrically isolate the system. This single, bidirectional data link also forces a half duplex mode, as well as tightly controlling communication through master polling and slaves responding, while being quiet for the rest of the time. There are some design alternatives to incorporate some semblance of interrupts, but most involve adding more lines to the system.
The real deciding factor is an ancillary feature related to the TWI and the microcontroller low power consumption “sleep” mode: the most consistent method of interrupting sleep mode or low power mode is with an interrupt generated by finding an address match on the TWI data line. Considering our stringent power consumption requirements, this is an invaluable feature.

RECOMMENDATIONS AND DECISIONS

Even considering the extra effort required to finalize a communication method on top of the existing protocol, the supplementary functions of using TWI are more than worth it when observed from a board function level. Research into isolating circuits built specifically for I²C should simplify the isolation challenge. Therefore, the focus of any additional innovation will lie in using this scheme for our master/slave communications.