

Differential bus communications for the Linear Technology's 6802 device

April 28, 2010

By Steven D. Letkeman, BSc. Zanthic Technologies Inc.

www.zanthic.com

Summary

This paper outlines a method of increasing communication reliability over an SPI (Serial Peripheral Interface) bus when used externally from a micro controller board by converting to a differential bus configuration. Specifically, this method is being used during prototype testing of the Linear Technology's 6802 multi cell measurement IC but the same methods could be applied to any SPI device. Although the 6802 is designed to communicate with a micro controller over the SPI bus, a normal method of deployment of this device would be to have the IC located off of the micro controller board in order to minimize the amount of high voltage leads being fed into the micro controller board. By creating smaller, remote boards, each monitoring 12 series cells, the amount of overall wiring and risk of high voltage shorts can be significantly reduced but would normally come at the cost of trying to get the SPI bus to work in an environment it was not intended for.

Solution

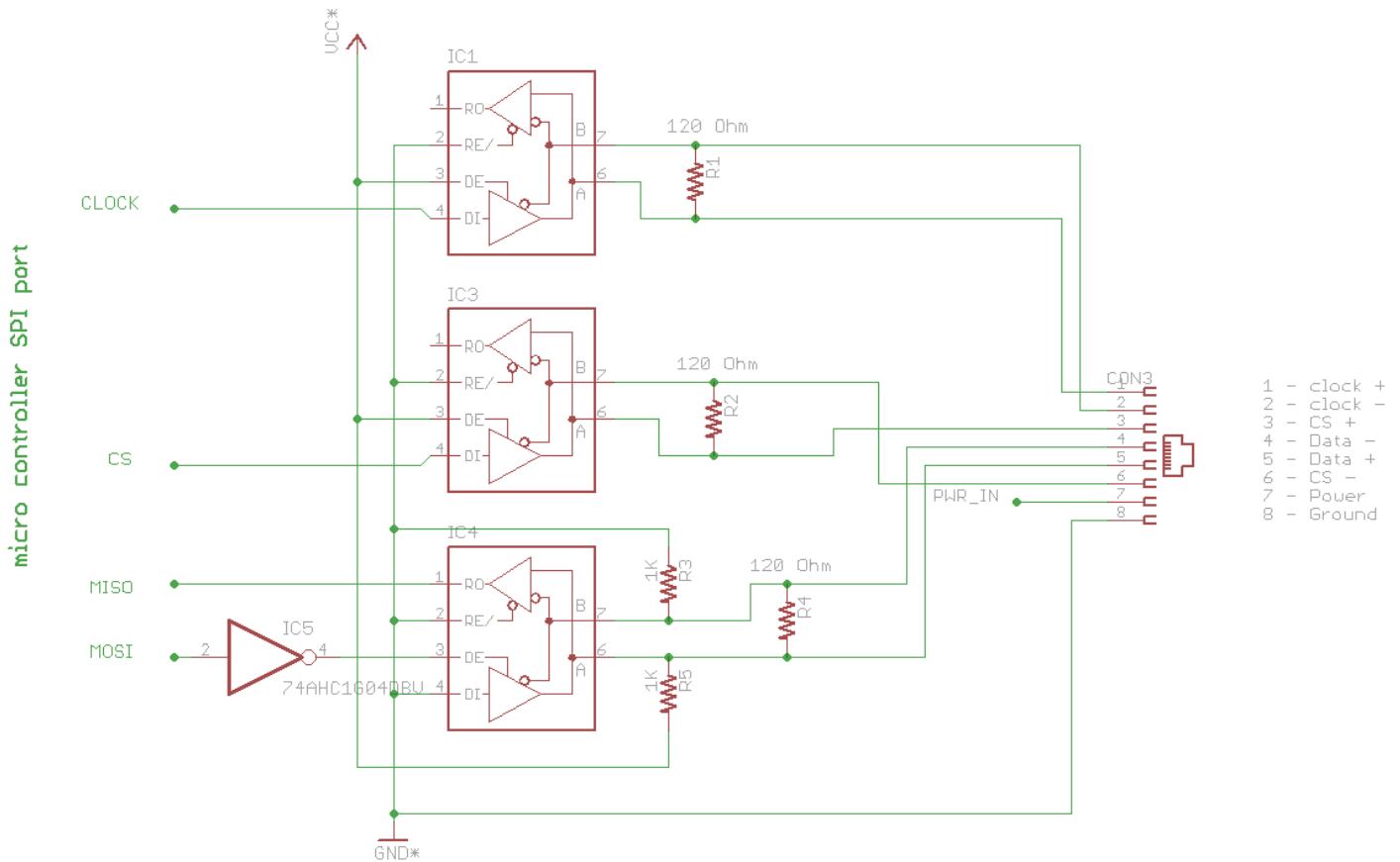
The schematics in this paper shows how a standard logic level SPI bus can be converted to differential signaling for a high noise immunity application such as high voltage, high current battery pack. A standard RJ45 Ethernet style cabling method is used with two connectors on each slave to allow for easy daisy chaining of slaves. The last slave on the chain can either get a special RJ45 plug with termination resistors or each board could have the termination resistors built in with jumpers to enable them. Three standard RS-485 driver IC's are used on each board. The RJ45 wiring is as follows

RJ45 Pin #	Description	Direction
1	Clock +	One direction, driven from uC
2	Clock -	
3	CS +	One direction, driven from uC
4	Data -	Bi-directional
5	Data +	
6	CS -	
7	DC Power	9 to 12 volts DC to power slaves
8	Ground (relative to uC)	uC ground

Standard RS-485 driver IC's were used for all transceivers even though only one of three on each board are actually bi-direction. This was done for the following reasons:

- 1) Available from many manufacturers.
- 2) Available in very low current consumption versions
- 3) Built in ESD protection
- 4) Low cost
- 5) Single part used

Master microcontroller SPI circuitry:



Schematic 1 – Microcontroller SPI driver

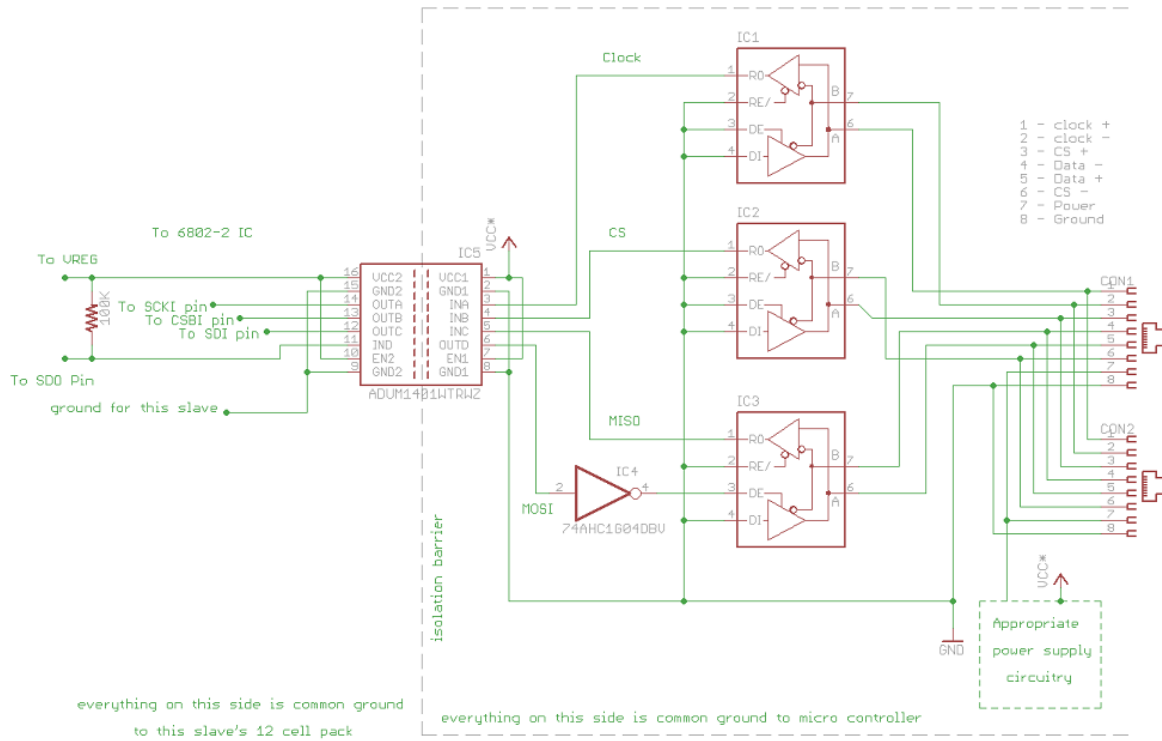
Notes:

- The clock and CS outputs are always enabled and therefore do not need the biasing resistors common for RS485 networks.
- The MOSI data being driven by the master goes to the enable circuitry on IC4, not the data input, which is tied to ground. The reason for this is that the RS-485 bus, when not driven, will be biased towards a high condition on the outputs. This means that we are only driving the low condition when transmitting data as we need to be able to also receive data from the slaves.

When the master transmits a low data bit, the IC5 inverter goes high, turning on the IC4 transmit enable and with the RS485 input tied low, drives a low out to the bus.

- This method of driving data has an implication to the master's method of data transfer, please see software notes.

Slave SPI circuitry:



Schematic 2 – Slave SPI receiver

Notes:

- A digital isolation barrier is shown in the slave schematic for use with the 6802 device but for other SPI applications it may not be necessary
- The Analog devices ADUM1401 device was chosen to ensure the 6802 VREG supply was not taxed beyond its 4mA capability
- With low current RS485 drivers, the total current draw on the driver side can be kept under 15mA per slave board
- The last slave should have 120 Ohm termination resistors across the clock, cs, and data lines

Software:

There is one critical change that must be made to the micro controller's code to allow this method to work. During a normal SPI transfer where the master is receiving data from the slave, the master will clock out some form of "dummy" data to the slave because, as the master is clocking out data, the slave is actually transmitting the real data. In the above scheme, where the master's data output is actually enabling the RS485 driver output when the master is outputting a low bit, the software must ensure that the dummy data is in fact all high bits. By transmitting high bits when expecting the slave's data, the master's RS485 driver is not enabled, allowing the slave's data to be received.

Conclusion

By using off the shelf RS-485 driver IC's, any SPI bus can be converted to a fully differential bus with higher noise immunity without sacrificing the convenience of SPI communications and can provide for reliable off board methods of wiring. For more information on further developments of this circuitry and firmware, please contact Zanthic Technologies Inc. If you are interested in developing your own 6802 slave circuitry for use with our battery monitoring system, please contact us for more information to better assist you.

About Steve Letkeman/Zanthic Technologies Inc.

Steve Letkeman is the owner of Zanthic Technologies Inc., which was incorporated in 1996. Steve has over 15 years of CAN (Controller Area Network) experience and over 22 years working with embedded microcontrollers. His designs can be found in everything from automotive, agricultural, robotic, home automation to industrial automation equipment. Zanthic products have been shipped to over 35 countries around the world.

For more information, please see our contact page at www.zanthic.com