



INNOVATIVE SENSOR TECHNOLOGY



# TSic™

## Precision Temperature Sensor IC

*Technical Notes – ZACwire™ Digital Output*

# IST TSic™ Temperature Sensor IC Technical Notes – ZACwire™ Digital Output

### CONTENTS

<b>1</b>	<b>ZACWIRE™ COMMUNICATION PROTOCOL FOR THE TSIC™</b> .....	<b>2</b>
1.1	TEMPERATURE TRANSMISSION PACKET FROM A TSIC™ .....	2
1.2	BIT ENCODING .....	3
1.3	HOW TO READ A PACKET .....	4
1.4	HOW TO READ A PACKET USING A μCONTROLLER .....	4

# TSic™

## Precision Temperature Sensor IC

### Technical Notes – ZACwire™ Digital Output

#### 1 ZACwire™ Communication Protocol for the TSic™

ZACwire™ is a single wire bi-directional communication protocol. The bit encoding is similar to Manchester in that clocking information is embedded into the signal (falling edges of the signal happen at regular periods). This allows the protocol to be largely insensitive to baud rate differences between the two ICs communicating. In end-user applications, the TSic™ will be transmitting temperature information and another IC in the system (most likely a  $\mu$ Controller) will be reading the temperature data over the ZACwire™.

##### 1.1 Temperature Transmission Packet from a TSic™

The TSic™ transmits 1-byte packets. These packets consist of a start bit, 8 data bits, and a parity bit. The nominal baud rate is 8kHz (125 $\mu$ sec bit window). The signal is normally high. When a transmission occurs, the start bit occurs first followed by the data bits (MSB first, LSB last). The packet ends with an even parity bit.

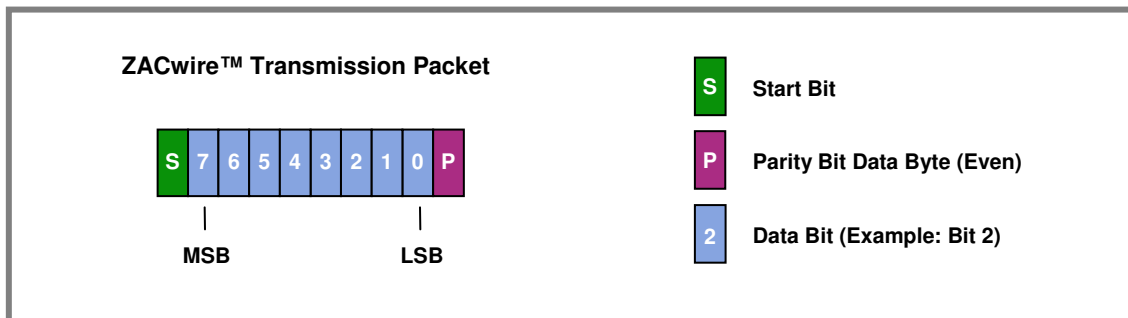


Figure 1.1 – ZACwire™ Transmission Packet

The TSic™ provides temperature data with 11-bit resolution,<sup>1</sup> which cannot be conveyed in a single packet. A complete temperature transmission from the TSic™ consists of two packets. The first packet contains the most significant 3 bits of temperature information, and the second packet contains the least significant 8 bits of temperature information. There is a single bit window of high signal (stop bit) between the end of the first transmission and the start of the second transmission.

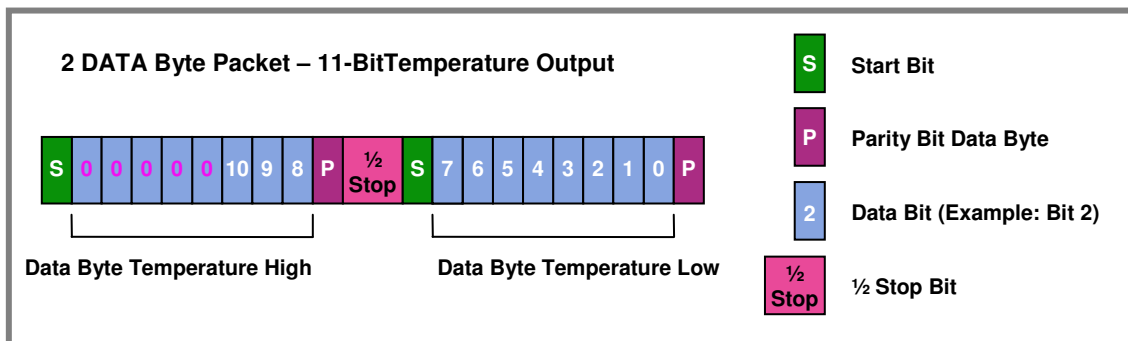


Figure 1.2 – Full ZACwire™ Temperature Transmission from TSic™ (5 MSBs are Zero-Padded)

<sup>1</sup> Contact ZMD for possible customization for higher temperature resolution.

# TSic™

## Precision Temperature Sensor IC

### Technical Notes – ZACwire™ Digital Output

#### 1.2 Bit Encoding

The bit format is duty cycle encoded:

Start bit => 50% duty cycle used to set up strobe time

Logic 1 => 75% duty cycle

Logic 0 => 25% duty cycle

Stop Bit

For the time of a half a bit width, the signal level is high. There is a half stop bit time between bytes in a packet.

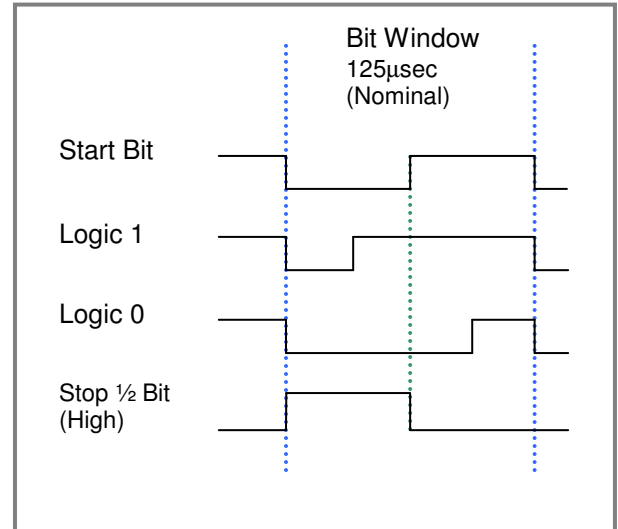


Figure 1.3 – Manchester Duty Cycle

An oscilloscope trace of a ZACwire™ transmission demonstrates the bit encoding. The following shows a single packet of 96Hex being transmitted. Because 96Hex is already even parity, the parity bit is zero.

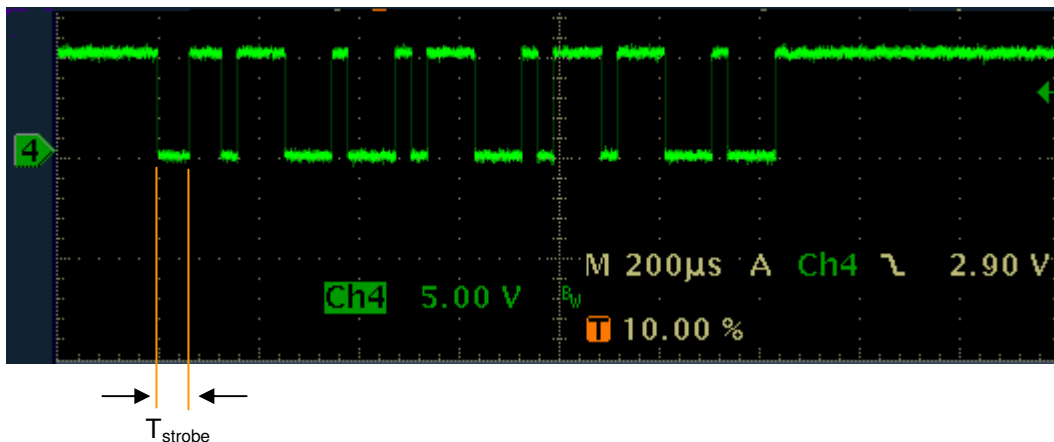


Figure 1.4 – ZACwire™ Transmission



# TSic™

## Precision Temperature Sensor IC

### Technical Notes – ZACwire™ Digital Output

#### 1.3 How to Read a Packet

When the falling edge of the start bit occurs, measure the time until the rising edge of the start bit. This time ( $T_{\text{strobe}}$ ) is the strobe time. When the next falling edge occurs, wait for a time period equal to  $T_{\text{strobe}}$ , and then sample the ZACwire™ signal. The data present on the signal at this time is the bit being transmitted. Because every bit starts with a falling edge, the sampling window is reset with every bit transmission. This means errors will not accrue for bits downstream from the start bit, as it would with a protocol such as RS232. It is recommended, however, that the sampling rate of the ZACwire™ signal when acquiring the start bit be at least 16x the nominal baud rate. Because the nominal baud rate is 8kHz, a minimum 128kHz sampling rate is recommended when acquiring  $T_{\text{strobe}}$ .

#### 1.4 How to Read a Packet using a $\mu$ Controller

It is best to connect the ZACwire™ signal to a pin on the  $\mu$ Controller that is capable of causing an interrupt on a falling edge. When the falling edge of the start bit occurs, it causes the  $\mu$ Controller to branch to its ISR. The ISR enters a counting loop incrementing a memory location ( $T_{\text{strobe}}$ ) until it sees a rise on the ZACwire™ signal. When  $T_{\text{strobe}}$  has been acquired, the ISR can simply wait for the next 9 falling edges (8 for data, 1 for parity). After each falling edge, it waits for  $T_{\text{strobe}}$  to expire and then samples the next bit.

The ZACwire™ line is driven by a strong CMOS push/pull driver. The parity bit is intended for use when the ZACwire™ is driving long (>2m) interconnects to the  $\mu$ Controller in a noisy environment. For systems in environments without noise interference, the user can choose to have the  $\mu$ Controller ignore the parity bit. Appendix A of this document gives an example of code for reading a TSic™ ZACwire™ transmission using a PIC16F627  $\mu$ Controller.

##### 1.4.1 How Often Does the TSic™ Transmit?

If the TSic™ is being read via an ISR, how often is it interrupting the  $\mu$ Controller with data? The update rate of the TSic™ is programmed to 10Hz (0.1ms response time). Servicing a temperature-read ISR requires about 2.7ms. Therefore the  $\mu$ Controller spends about 2.7% of its time reading the temperature transmissions.

# TSic™

## Precision Temperature Sensor IC

### Technical Notes – ZACwire™ Digital Output

#### 1.4.2 Solutions if a Real Time System Cannot Tolerate the TSic™ Interrupting the $\mu$ Controller

Some real time systems cannot tolerate the TSic™ interrupting the  $\mu$ Controller. In this case, the  $\mu$ Controller must initiate the temperature read. This can be accomplished by using another pin of the  $\mu$ Controller to supply VDD to the TSic™. The TSic™ will transmit its first temperature reading approximately 65ms to 85ms after power up. When it is time for the  $\mu$ Controller to read the temperature, it first powers the TSic™ using one of its port pins. It will receive a temperature transmission approximately 65ms to 85ms later. If during that time, a higher priority interrupt occurs, the  $\mu$ Controller can simply power down the TSic™ to ensure it will not cause an interrupt or be in the middle of a transmission when the higher priority ISR finishes. This method of powering the TSic™ has the additional benefit of acting like a power down mode and reducing the quiescent current from a nominal 45 $\mu$ A to zero. The TSic™ is a mixed signal IC and provides best performance with a low-noise VDD supply. Powering through a  $\mu$ Controller pin does subject it to the digital noise present on the  $\mu$ Controller's power supply. Therefore it is best to use a simple RC filter when powering the TSic™ with a  $\mu$ Controller port pin. See the diagram below.

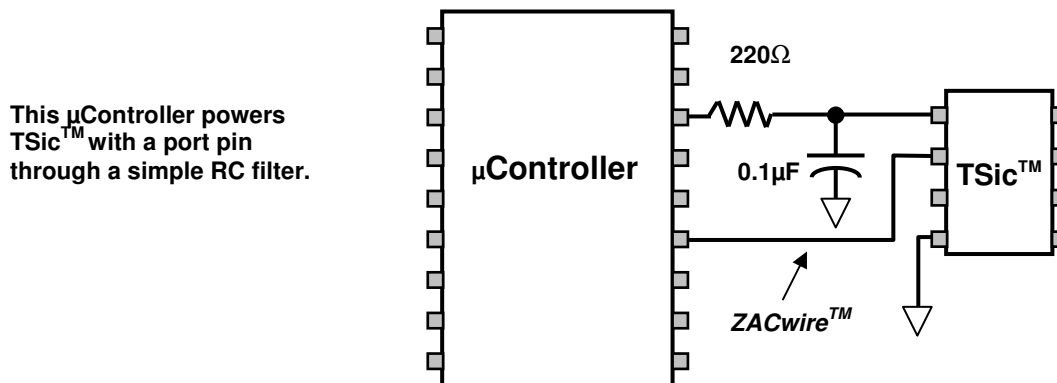


Figure 1.5 – RC Filter for Powering TSic™ through the  $\mu$ Controller