Application Report

Interfacing TMP107 SMAART wire™ Temperature Sensors Over CAN or RS-485 Physical Layer

Jiri Panacek *Automotive Systems*

ABSTRACT

The TMP104, TMP107, TMP107-Q1, or TMP144 devices are temperature sensors that feature the SMAART wire™ interface. The interface uses asynchronous, half-duplex, transfer compatible with most UART peripheries. Additionally, it supports daisy-chain mode of operation for multiple sensors on the same bus. Remote sensors in automotive, industrial or home automation often require a standardized physical layer such as RS-485 or CAN. This application report describes a simple, low-cost, circuit that allows interfacing SMAART wire™ sensors over RS-485 and CAN bus. The TMP107-Q1 serves as an example.

Table of Contents

Trademarks

SMAART wire™, C2000™, LaunchPad™, are trademarks of Texas Instruments.
All trademarks are the property of their respective owners.

1 Introduction

The SMAART wire bus interface communicates over a single wire using a half-duplex UART (Universal Asynchronous Receiver Transmitter) interface. The half-duplex communication requires careful timing and mechanisms that prevent collisions on the bus. The auto-baud detection feature ensures that the temperature sensor always replies at the same communication speed (baud rate).

Figure 1-1 shows a snippet of the communication for the *Address initialize* command. The temperature sensor listens to the bus by default. Its input pin I/O is high impedance. The master device always initiates the communication first. The temperature sensor uses the first calibration byte 0x55 for baud rate detection. The second byte 0x95 corresponds to the *Address initialize* command. The third byte 0x85 assigns the first device address to 8. Then, the master device releases the bus and waits for the reply from the sensor. Each temperature sensor on the SMAART wire bus interface replies with the assigned 5-bit address (bytes 0x83, 0x8B). See the *Communication Protocol* section in the *TMP107-Q1 Automotive Grade, ±0.4°C Temperature Sensor with Daisy-Chain UART, EEPROM, and Alert Function Data Sheet*, for details.

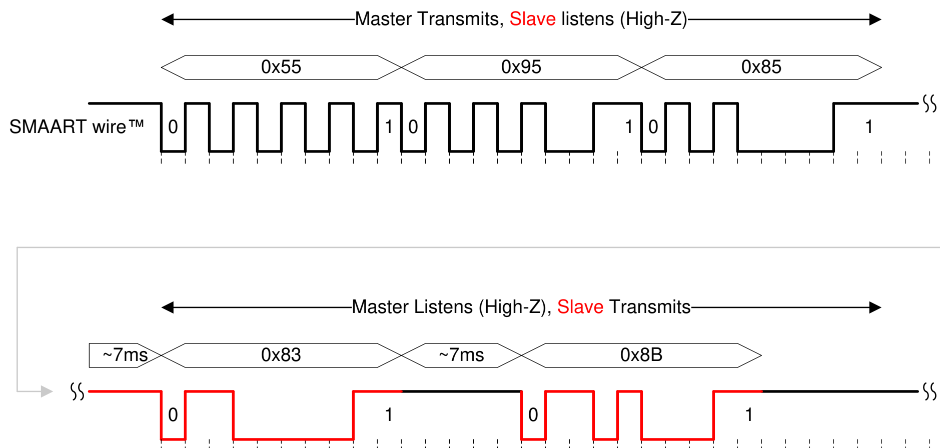


Figure 1-1. TMP107 Communication Example - Identify Command

Figure 1-2 shows an example of interfacing the TMP107-Q1 temperature sensor. The TXD signal drives the enable pin of the buffer instead of driving the input. This implementation converts the push-pull output to open drain (open collector). The 4.7-kΩ pullup resistor brings the bus to logic 1 during standby. The SMAART wire bus interface transfers data reliably up to 300 meters. This also means that the parasitic inductance of the signal wire and mismatched impedance may cause ringing and overshoots. The 100-Ω resistor protects the RXD input from damage by reducing the current through the integrated ESD protection diodes during transient events below GND and above VCC rail. The 10-Ω resistor eases signal collisions and slows down signal edges. This improves signal integrity.

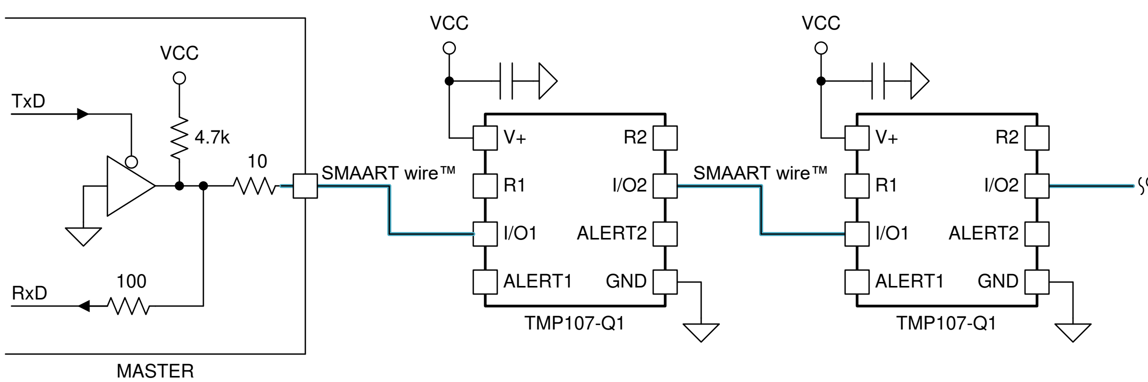


Figure 1-2. TMP107 SMAART wire™ Interface

2 CAN and RS-485 Physical Layers

Although the SMAART wire bus interface is proven to be reliable, both the CAN and RS-485 physical layers offer a significant improvement in noisy environments due to differential signaling. The difference between the CAN and RS-485 transceiver is the way they drive the bus.

The CAN transceiver uses an open collector output and does not require any driver enable signal. Multiple transceivers transmitting at the same time ruin data integrity but there is not any excessive current flowing through the bus. The CAN bus defines a recessive and dominant state on the bus. These states represent logic values zero or one. Figure 2-1 shows the fundamentals. See the blog article [1] for further details.

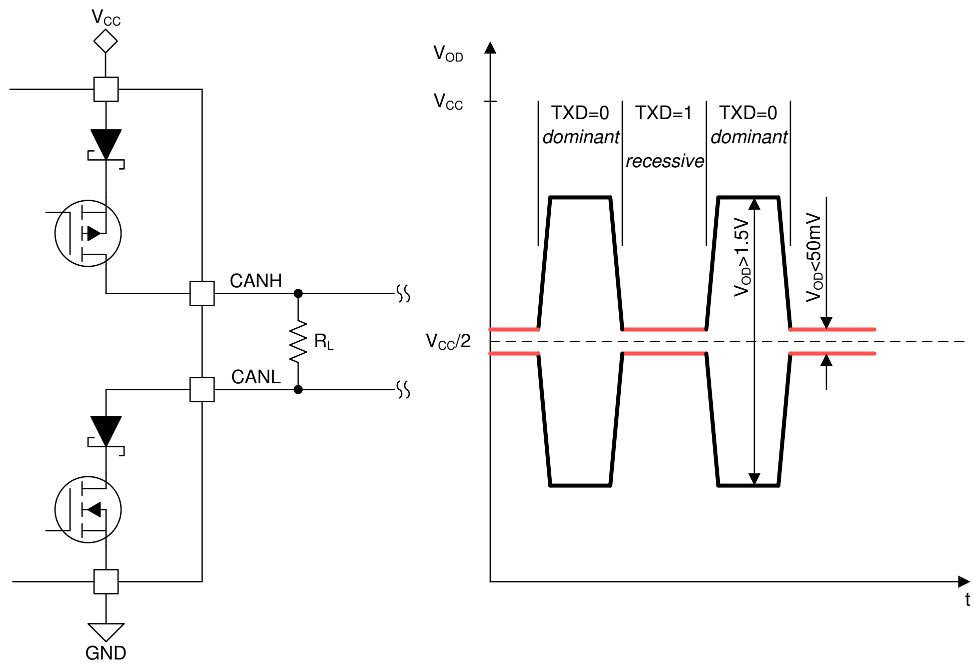


Figure 2-1. Fundamentals of the CAN bus

The RS-485 transceiver uses a push-pull configuration and requires a driver enable (DE) and receive enable (~RE) signal for direction control. The bus driver either actively drives the bus or enters the idle mode. In the idle mode the driver represents high impedance and external or internal fail-safe biasing ensures a default logic value of the bus. Any active driver can override the bus in the idle mode. In the active mode the push-pull driver controls nodes A and B. The differential voltage between the pins is positive or negative. The polarity of the differential voltage represents logic values on the bus. Higher voltage on the node A than on B represents logic 1. Lower voltage on the node A than B represents logic 0.

Multiple RS-485 transceivers transmitting at the same time cause bus contention. One transceiver may force the signal line high whereas the second forces it low. Modern transceivers use protections but this unwanted event causes excessive power dissipation and ruins data integrity. Figure 2-2 shows fundamentals of the RS-485 bus. See the blog [2] for additional information on RS-485 bus.

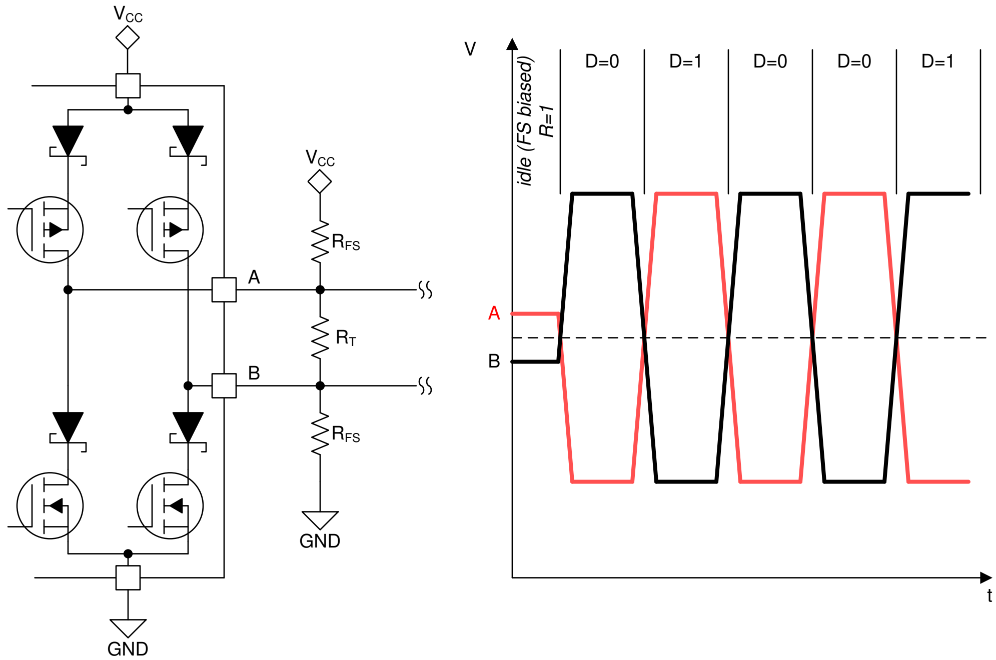


Figure 2-2. Fundamentals of the RS-485 bus

3 TMP107-Q1 Over the CAN Bus

The CAN transceiver has dedicated pins for receiving (RXD) and transmitting (TXD) data. The TMP107-Q1 sensor has a single bidirectional pin (I/O). Connecting the bidirectional pin I/O directly to the RXD and TXD pins is not possible. The CAN transceiver has an internal loopback that mirrors all data on the CAN bus to the RXD pin. Merging RXD and TXD pins together would result in the bus conflict with undefined behavior.

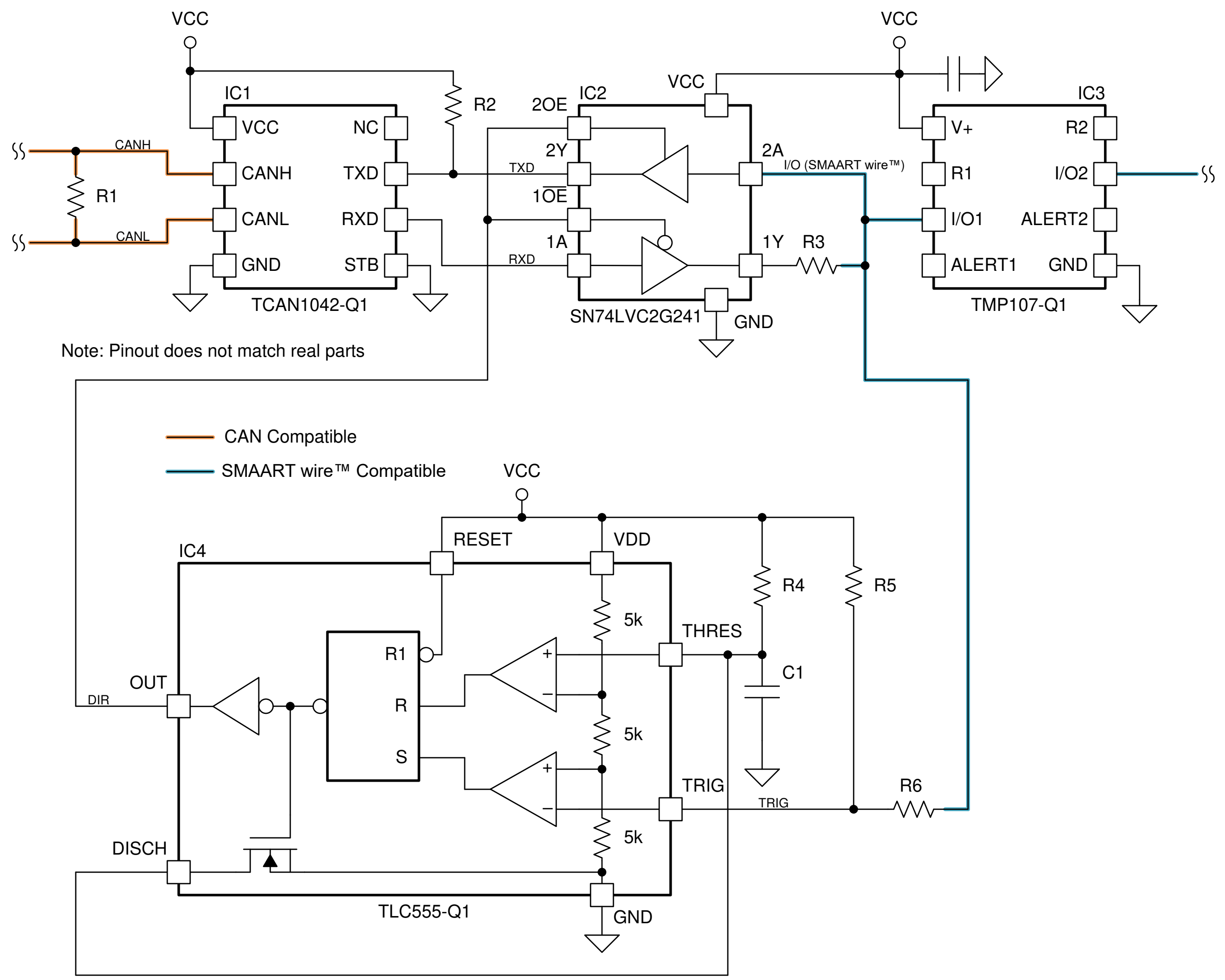


Figure 3-1. Simplified Circuit Diagram for Interfacing the TMP107-Q1 Over the CAN bus

Figure 3-1 shows a circuit that allows interfacing the TMP107-Q1 sensor over the CAN bus. Two logic buffers in IC2 convert the single bidirectional I/O signal to unidirectional RXD and TXD signals. Buffer enable pins $1\overline{OE}$ and 2OE are complementary and share the same input signal (DIR). This means only one buffer is active at the same time. This functionality allows for data direction control and prevents the bus conflict due to the loopback mechanism in the CAN transceiver IC1. When the DIR signal is low, the TMP107-Q1 sensor IC3 listens to the CAN bus. When the DIR signal is high, the sensor transmits data to the master device over the CAN bus.

The TMP107-Q1 device does not have any dedicated pin that indicates receiving or transmitting state. An external circuit detects when the sensor needs to send data and toggles the buffer IC2 accordingly. The external circuit uses the industry standard 555 timer. Alternatively an analog comparator can perform the same functionality.

The detection mechanism uses different voltage levels for the logic 0 on the SMAART wire bus interface. Only the logic 0 is important because this corresponds to the dominant state on the CAN bus.

Logic zero from the CAN bus to the SMAART wire™ bus interface

Careful selection of resistors R3, R5, R6 ensures that a logic zero on the 1Y pin of IC2 never causes voltage on the TRIG pin of the IC4 reaching the triggering threshold VCC/3. Additionally, a proper combination of resistors R3, R6 sets the voltage on the SMAART wire bus interface below the low-level input voltage (VIL) threshold. This way the sensor still detects logic 0.

Logic zero from the SMAART wire™ bus interface to the CAN bus

The TMP107-Q1 device asserts logic 0. This factors out the resistor R3 and voltage on the TRIG pin reaches the triggering threshold VCC/3. The OUT pin of the IC4 toggles high and enables the buffer IC2 for transmission. Without the optional capacitor C1 the output remains high only as long as the TRIG signal is below the threshold. This means that the pull-up resistor R2 maintains logic 1 on the TXD pin. The capacitor C1 can extend the OUT pulse duration for the complete frame. This is beneficial for higher transfer speeds when the pull-up resistor R2 may not be able to quickly bring the TXD pin to logic 1.

Figure 3-2 shows example waveforms for the CAN to SMAART wire interface converter circuit from Figure 3-1.

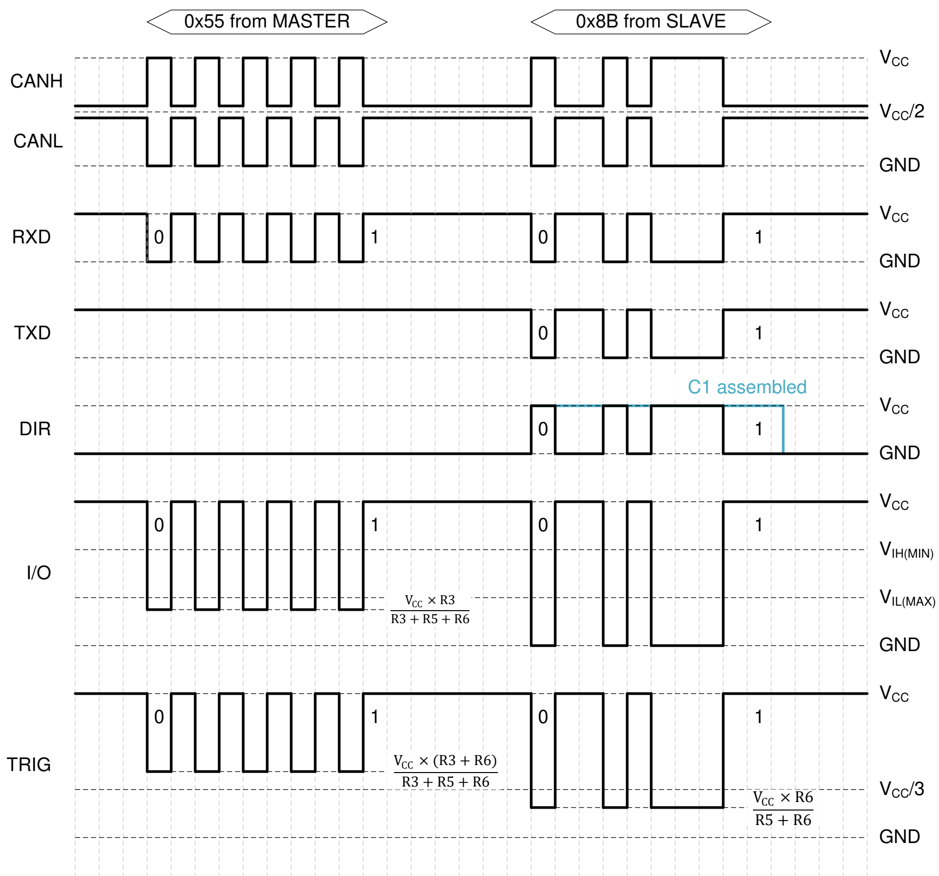


Figure 3-2. Ideal Waveforms for the CAN to SMAART wire™ Interface Converter Circuit (Example Only)

4 TMP107-Q1 Over the RS-485 Bus

Figure 4-1 shows the implementation with the RS-485 transceiver. The transceiver features complementary enable signals for the transmitter and the receiver. This integrated functionality replaces the SN74LVC2G241 buffer from the CAN-based circuit. For this reason, the DIR signal from the TLC555-Q1 timer connects directly to the DE and ~RE signals.

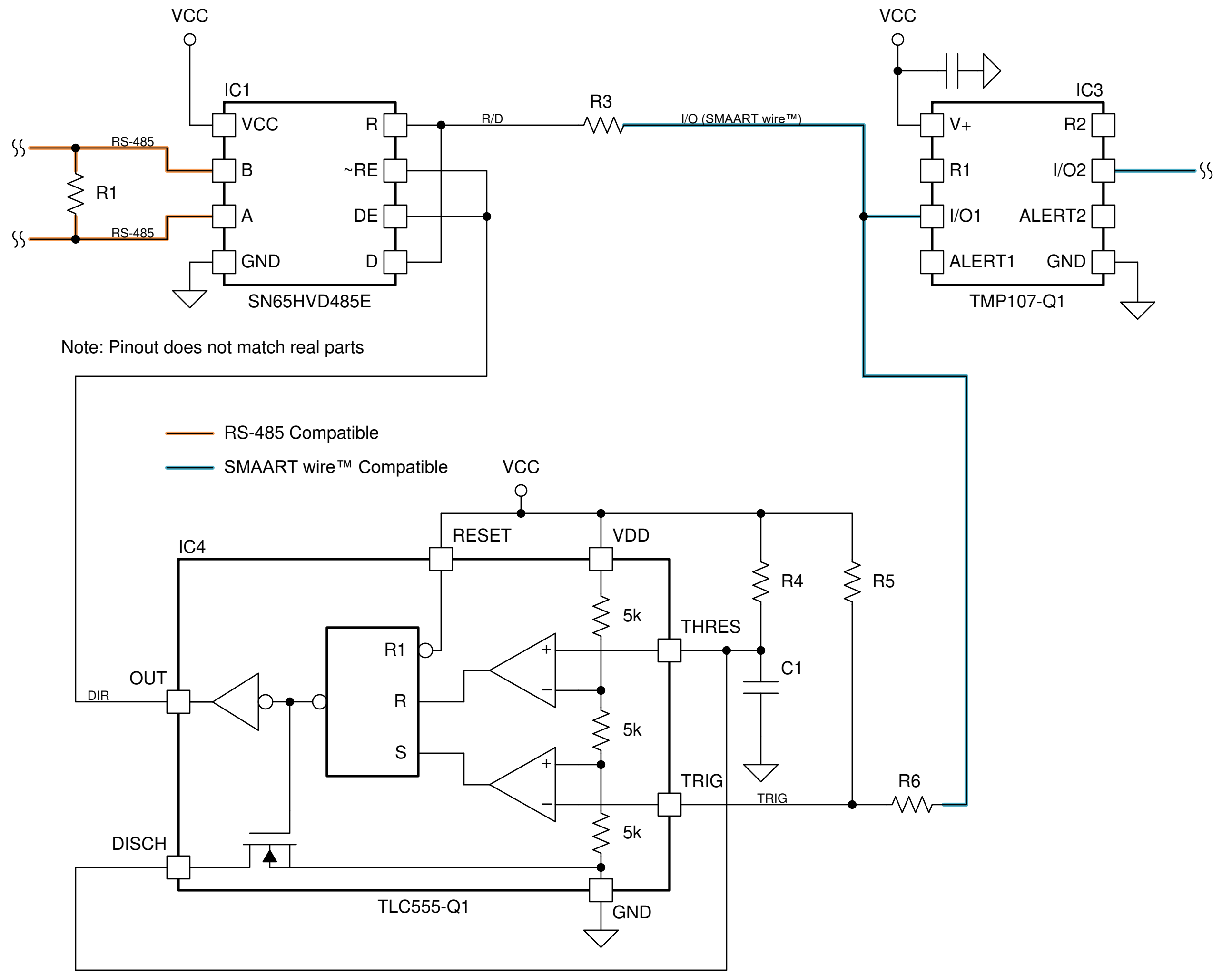


Figure 4-1. Simplified Circuit Diagram for Interfacing the TMP107-Q1 Over the RS-485 bus

Figure 4-2 represents ideal waveforms for the circuit with the RS-485 transceiver.

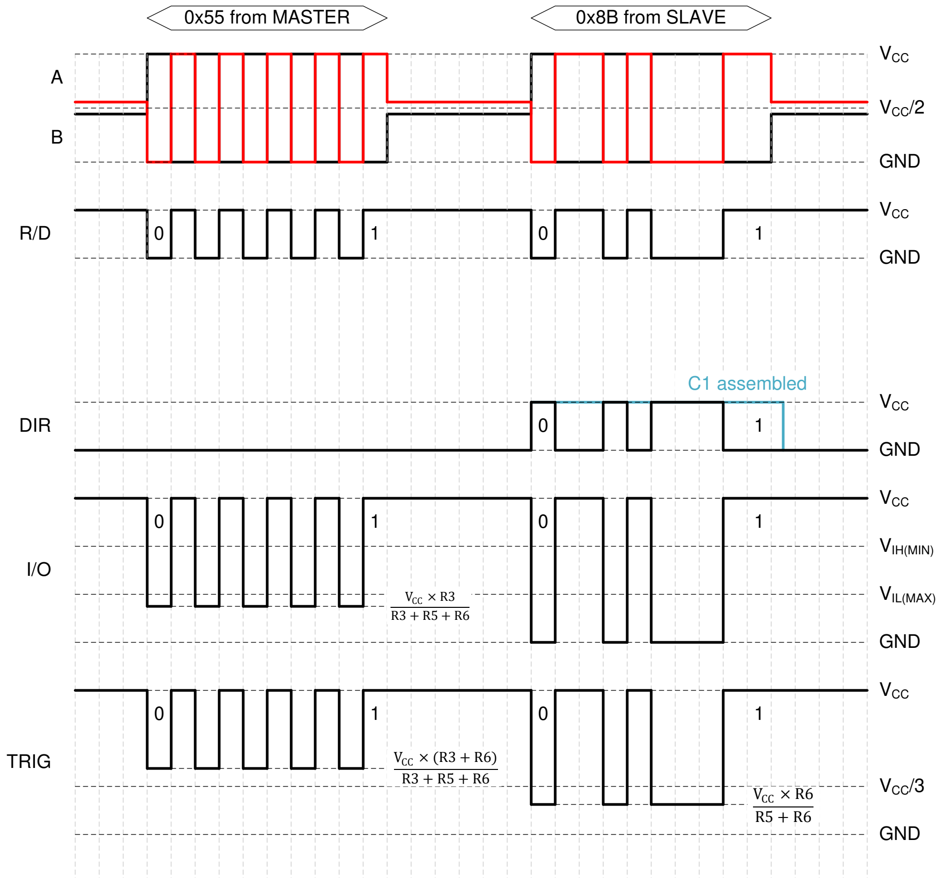


Figure 4-2. Ideal Waveforms for the RS-485 to SMAART wire™ Interface Converter Circuit (Example Only)

5 Test Setup

The test setup consists of the LAUNCHXL-F280049C development board and custom PCB boards. Figure 5-1 and Figure 5-2 show the actual circuit diagrams for boards under test. Figure 5-3 shows the test setup. Small adapter boards plugged directly to the C2000™ MCU LaunchPad™ convert UART interface to CAN or RS-485. Small converters in the middle translate the signal to SMAART wire bus interface compatible. Printed circuit boards on the right host individual TMP107-Q1 temperature sensors.

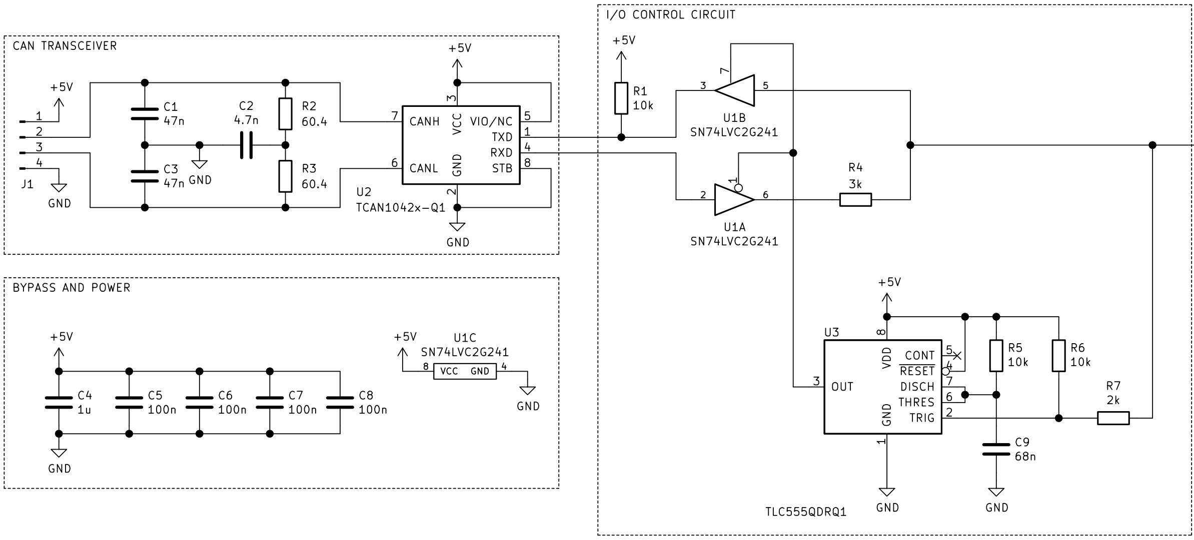


Figure 5-1. Circuit Diagram of the CAN to SMAART wire™ Interface Converter

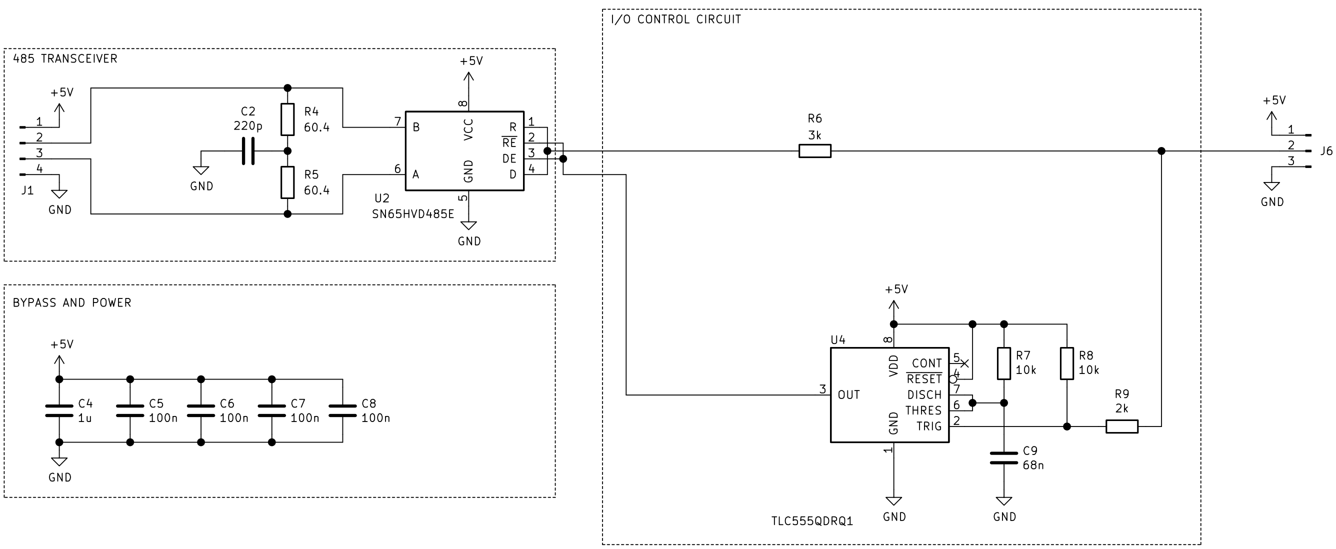


Figure 5-2. Circuit Diagram of the RS-485 to SMAART wire™ Interface Converter

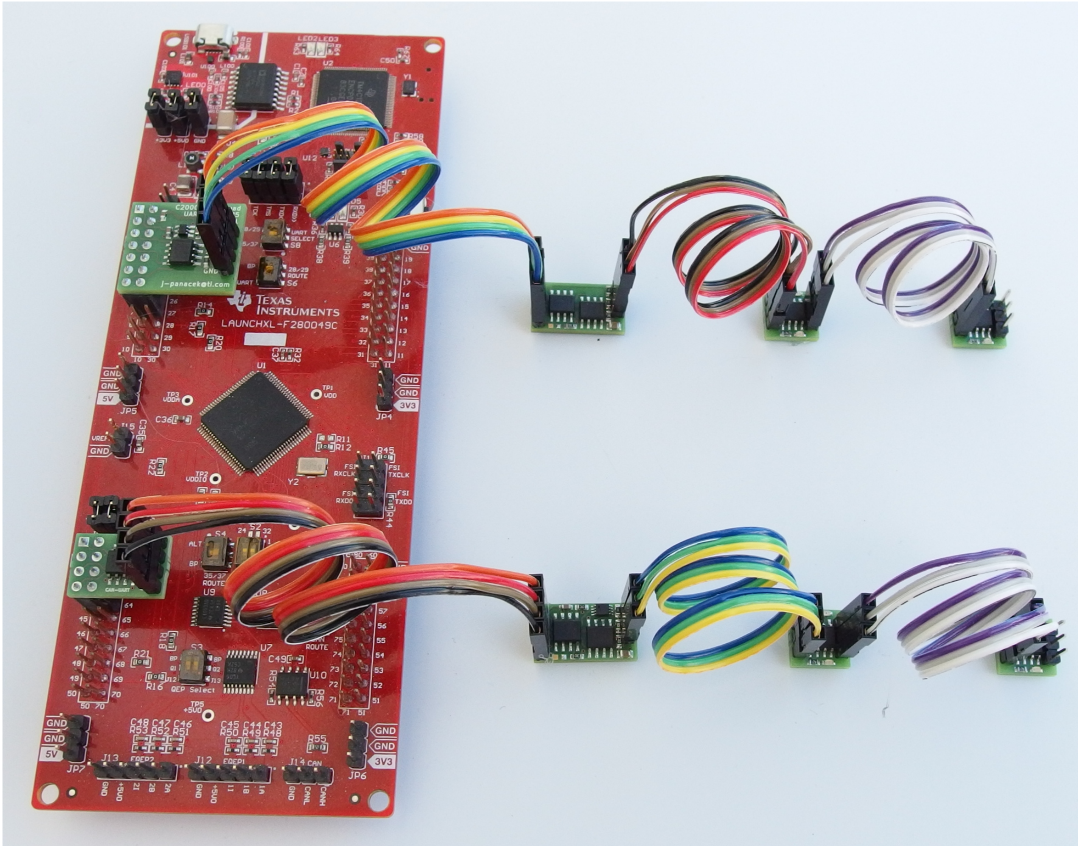


Figure 5-3. Test Setup

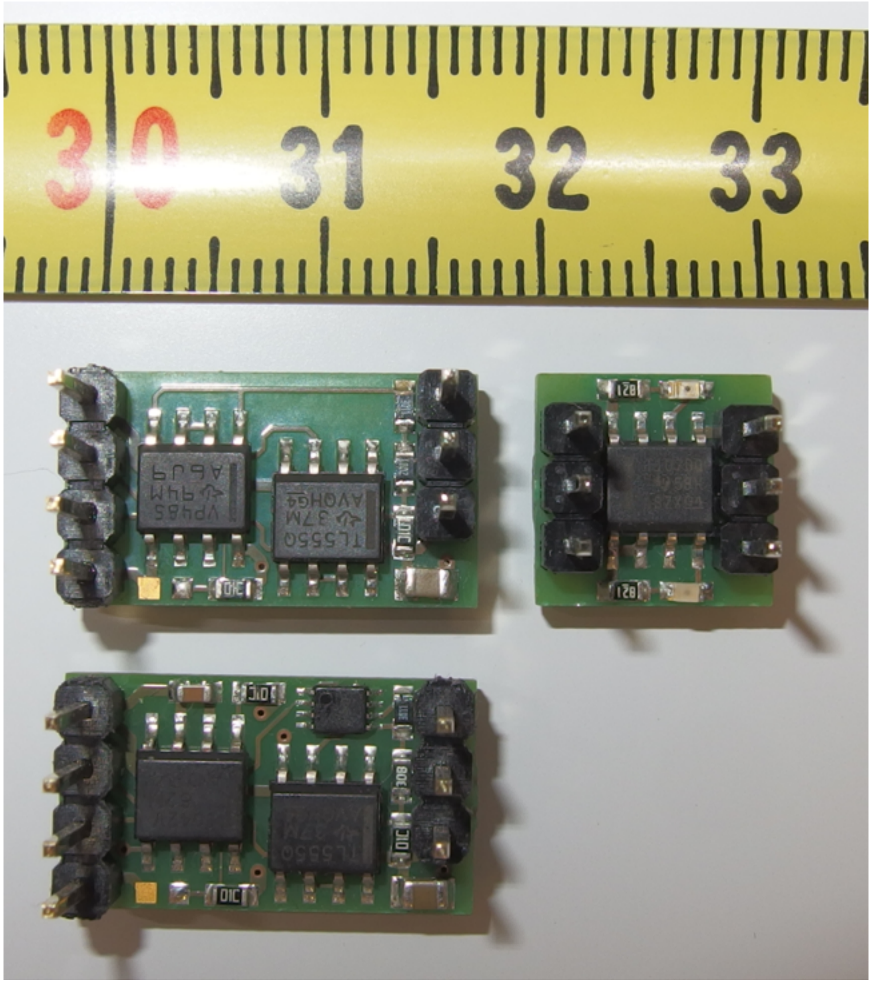


Figure 5-4. Detail of the Converter Boards and the Temperature Sensor Board

---

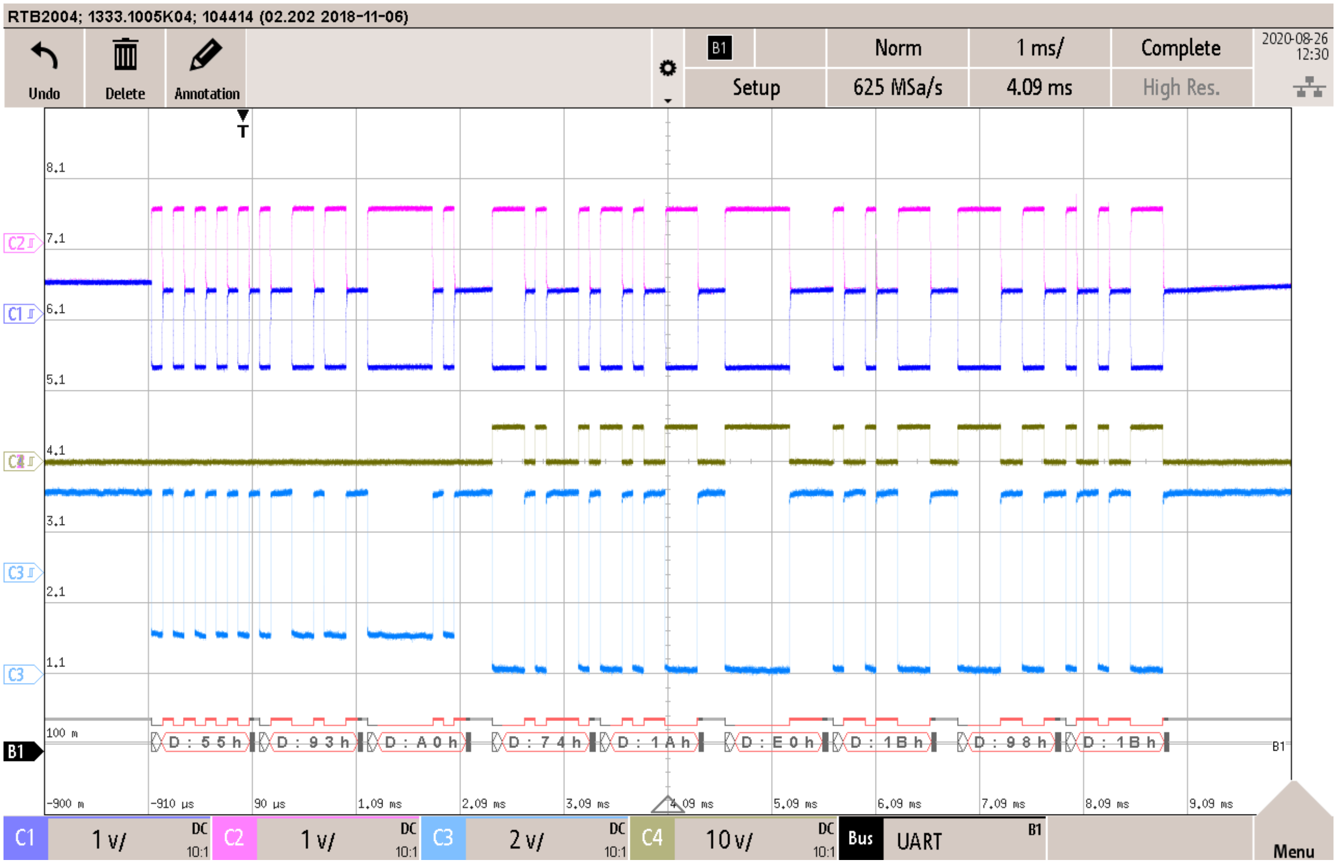
Note

Contact TI on the www.ti.com/e2e support forum for further information on the demo firmware for the C2000™ microcontroller and design data in KiCad format.

---

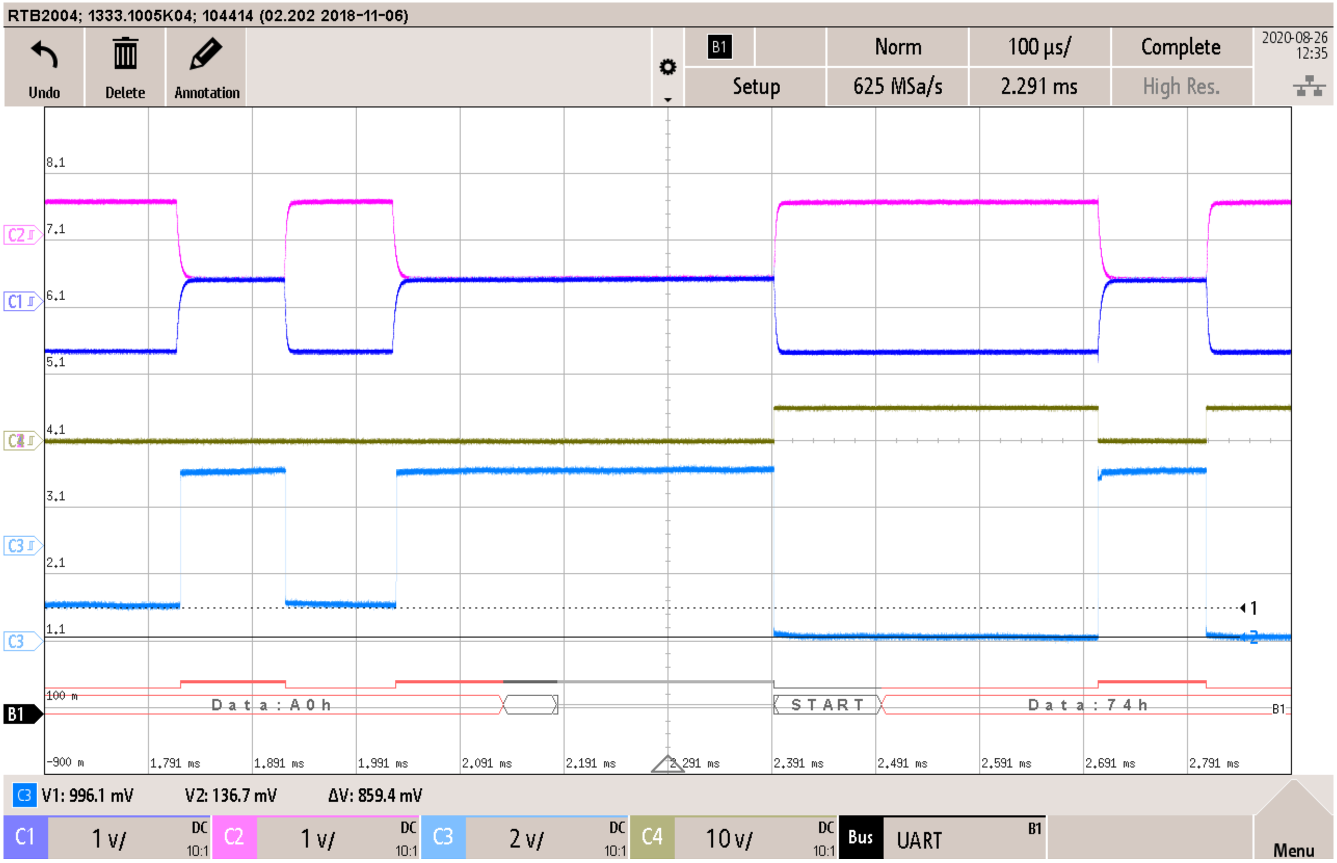
6 Test Results

Figure 6-1 and Figure 6-2 show real waveforms for the CAN to SMAART wire interface converter circuit. The host microcontroller reads out temperature from three sensors at 9600 bps.



Channel1: CANL, Channel2: CANH
Channel3: I/O, Channel4: DIR

Figure 6-1. Real Waveforms for the CAN to SMAART wire™ Interface Converter Circuit



Channel1: CANL, Channel2: CANH
Channel3: I/O, Channel4: DIR

Figure 6-2. Real Waveforms for the CAN to SMAART wire™ Interface Converter Circuit

Figure 6-3 shows real waveforms for the RS-485 to SMAART wire interface converter without the timing capacitor C9. Note that during the TMP107-Q1 sensor transmitting the RS-485 bus switches only between the logic-low and idle state. The communication works because the receive data output pin R of the SN65HVC485 transceiver on the receiving side returns logic high for the properly-terminated bus in the idle state.

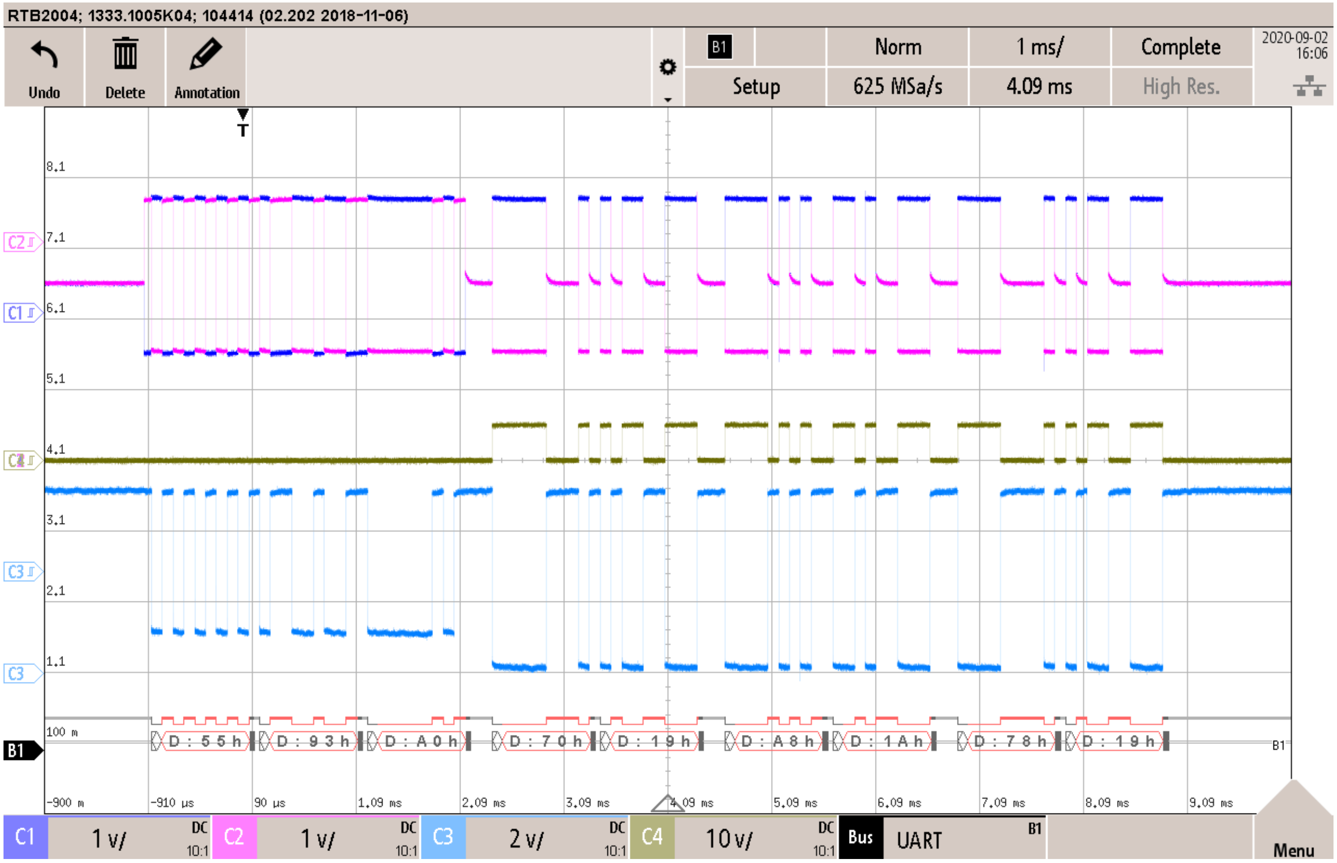


Figure 6-3. Real Waveforms for the RS-485 to SMAART wire™ Interface Converter Circuit Without the Timing Capacitor C8

Adding the capacitor C8 (68 nF) extends the pulse and forces the RS-485 bus remaining in the active state for longer time as shown in the Figure 6-4. The control signal DIR returns to zero during transmitting the stop bit over the bus.

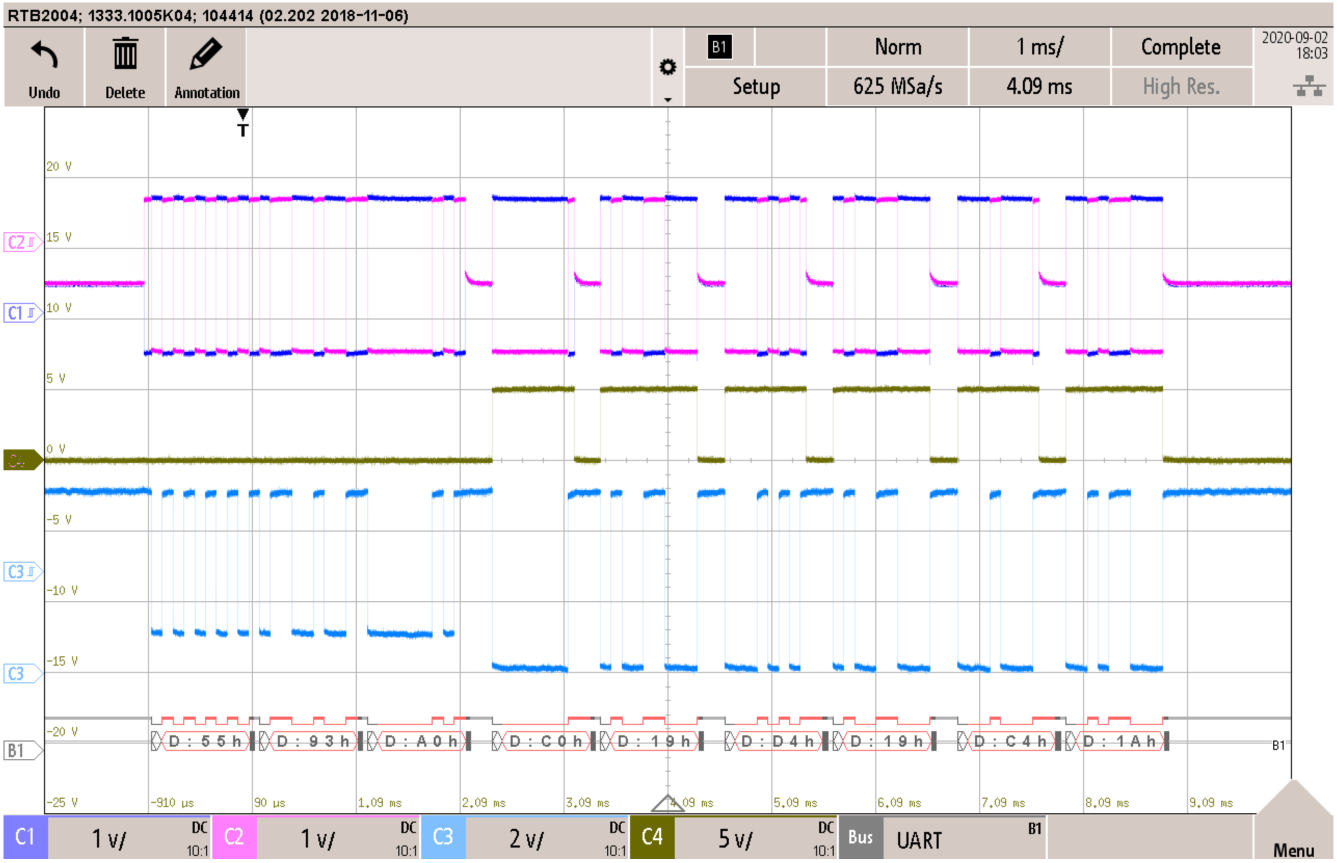


Figure 6-4. Real Waveforms for the RS-485 to SMAART wire™ Interface Converter Circuit With the Timing Capacitor C8 (68 nF)

Figure 6-5 and Figure 6-6 show further improvement with the timing capacitor C8 and the diode D1. The diode from the timing capacitor to the I/O signal converts the circuit into re-triggable single-shot timer. At least every start bit re-triggers the timing circuitry and extends the period for which the DIR signal keeps the bus in active state.

Figure 6-5. Real Waveforms for the RS-485 to SMAART wire™ Interface Converter Circuit With the Timing Capacitor C8 and Diode D1

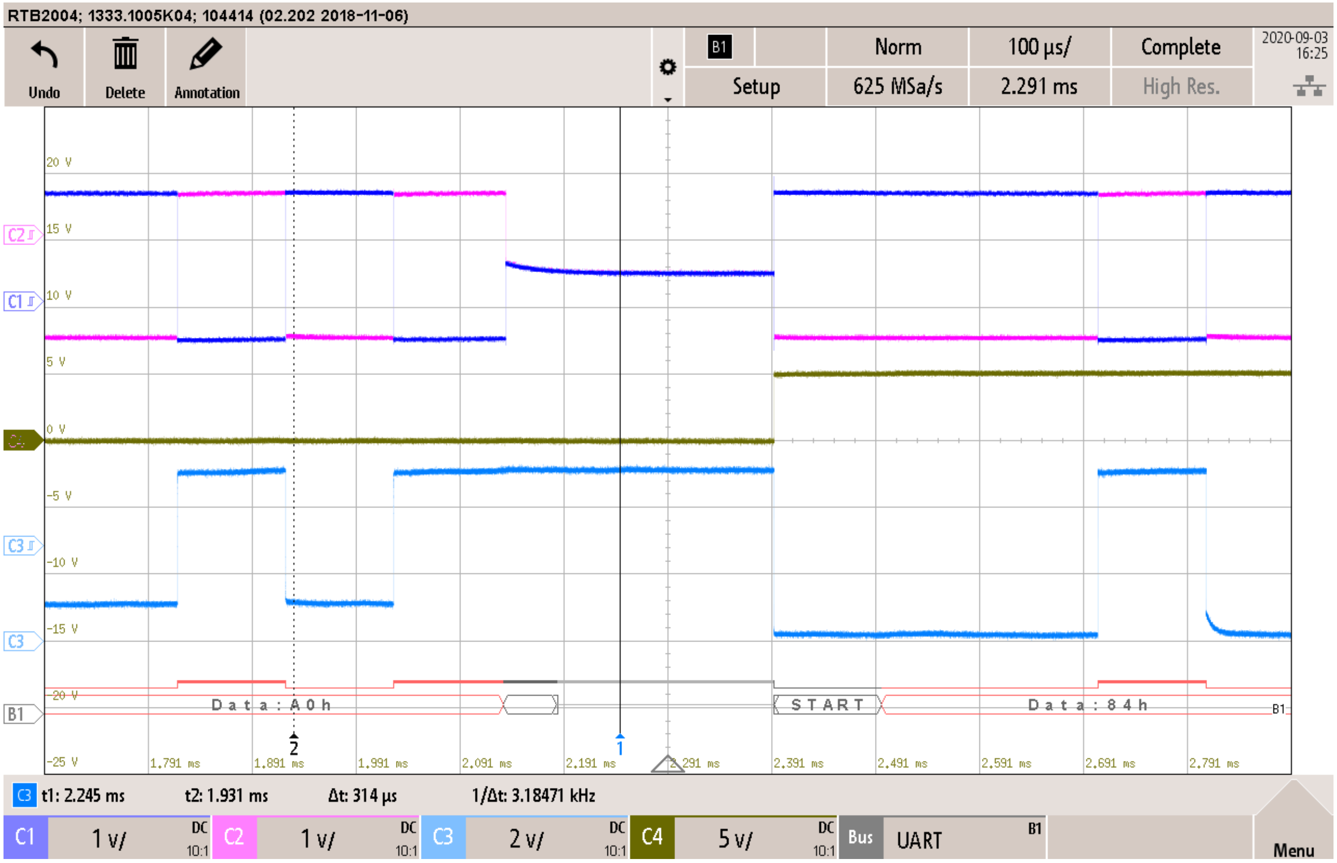


Figure 6-6. Real Waveforms for the RS-485 to SMAART wire™ Interface Converter Circuit With the Timing Capacitor C8 and Diode D1 (Detail)

7 Conclusion

This application reports shows how to interface the TMP107-Q1 temperature sensor, or similar, over the standardized CAN or RS-485 physical layer. The circuit uses an inexpensive TLC555-Q1 timer with integrated comparators for the flow control. This solution eliminates the microcontroller and allows for long-distance communication in automotive and industrial applications.

8 References

1. *Learn the inner workings of a CAN bus driver and how to debug your system*, Blog, available online from https://e2e.ti.com/blogs_/b/industrial_strength/archive/2016/01/26/the-inner-workings-of-a-can-bus-driver
2. *RS-485 basics technical article series*, Blog, available online from http://e2e.ti.com/blogs_/b/industrial_strength/archive/2015/06/16/rs-485-basics-the-rs-485-driver