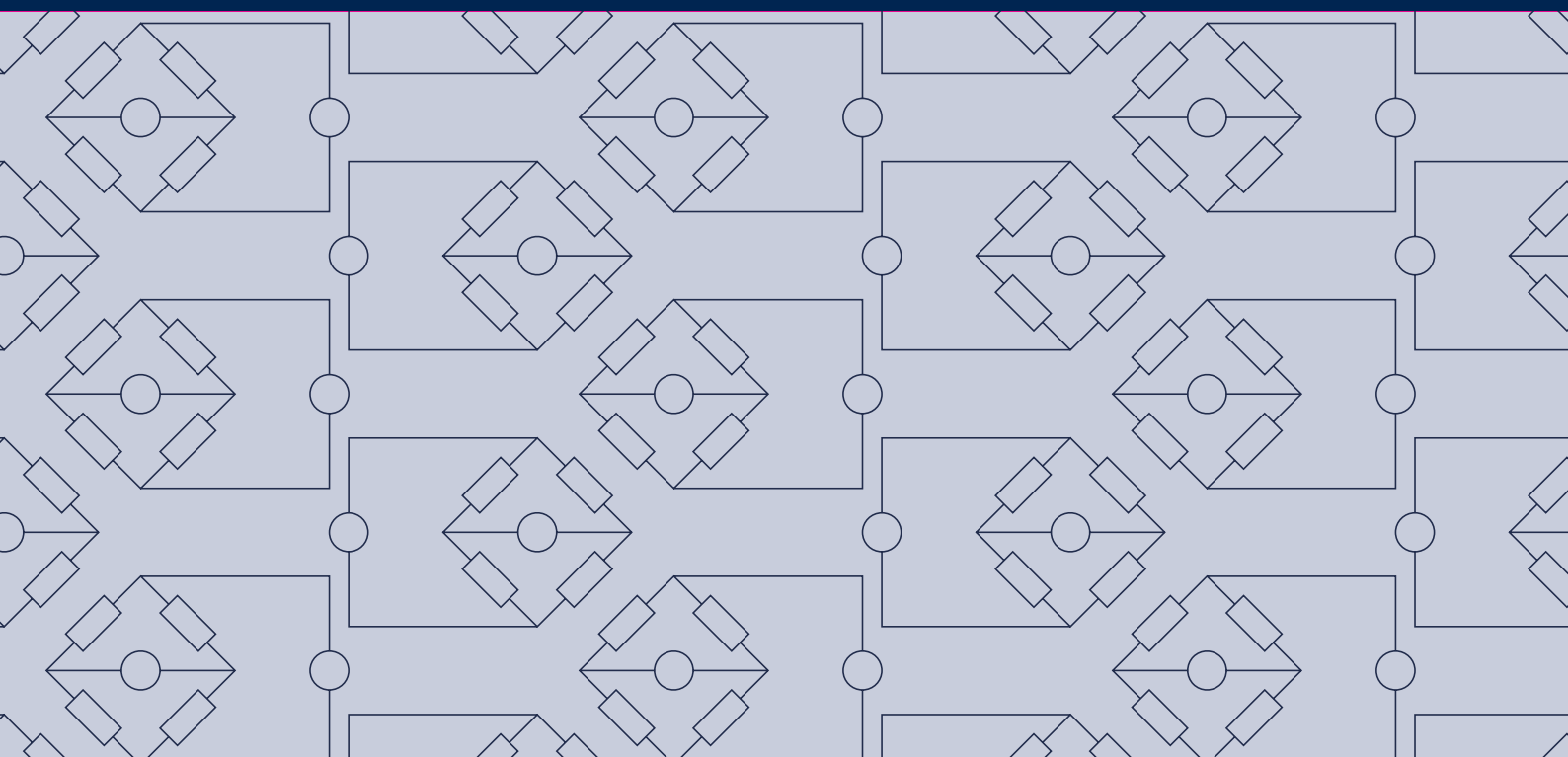
First Sensor is now part of TE connectivity

SPI bus communication with LDE/LME pressure sensors

Application note



First Sensor is now part of TE connectivity

SPI bus communication with LDE/LME pressure sensors

This Application Note discusses methods and special considerations related to the Serial Peripheral Interface (SPI) protocol used to communicate digitally with LDE and LME series pressure sensors.

1. Scope

It is outside of the scope of this document to define the SPI protocol itself; many excellent resources on the topic are readily available for reference[1].

It is also outside of the scope of this document to discuss the use of LDE devices for analog output and/or as drop-in replacements for LBA devices. Please refer to the LDE datasheet for more information on this.

2. LDE/LME device overview

2.1 Basic overview of device operation

LDE and LME series low differential pressure sensors work on the thermal-anemometer principle, where a heating element is bounded by a symmetrical arrangement of upstream and downstream temperature sensors. A differential pressure applied across the ports of the device creates a flow across this sensing element. As pressure (and thus flow) increases, the temperature profile of the heater shifts closer to one or the other temperature sensor; this difference between the upstream and downstream pressure can therefore be extrapolated to the applied differential pressure.

This sensing element's voltage output is acquired and converted to a digital signal by the embedded microcontroller's ADC. Processing is done to linearize the output and tare offset, and a digital output is then passed to either a DAC to provide an analog output signal, or to the serial interface for transfer over the SPI bus.

2.2 Hardware block diagrams/explanation of differences

While the overview of the device in the previous paragraph is sufficient to understand how the sensor works, there are subtle differences in the underlying architecture of the LDE/LME series sensors, depending on their supply voltage.

[1] **Useful reference materials include:**

1. AN991, "Using the Serial Peripheral Interface to Communicate Between Multiple Microcomputers", Motorola Inc., 1987.
2. "Serial Peripheral Interface Bus", Wikipedia (http://en.wikipedia.org/wiki/Serial_Peripheral_Interface_Bus), accessed 17 September, 2014.
3. The manufacturer's datasheet and any relevant application notes for the SPI master device intended to be used.

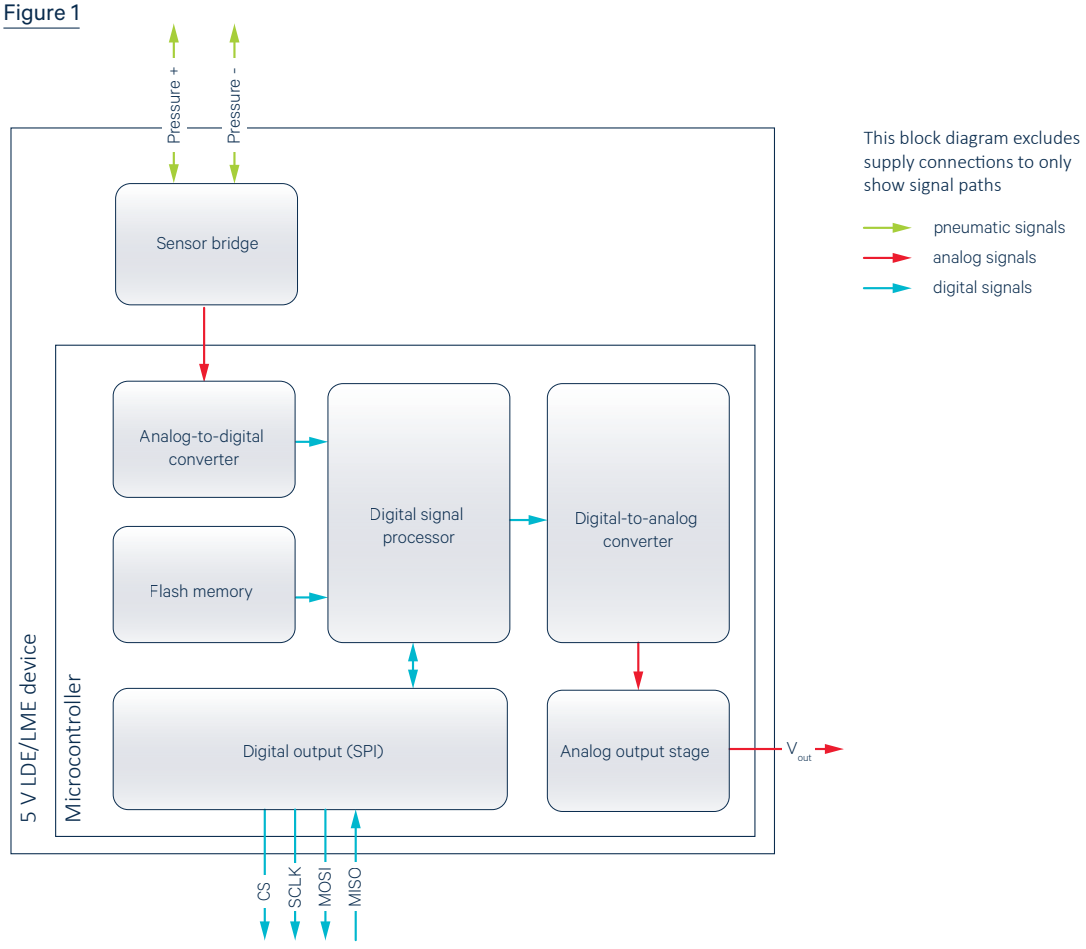
SPI bus communication with LDE/LME pressure sensors

2.2.1 5 V LDE/LME devices

At its essence, the LDE/LME devices consist of a sensor bridge and a microcontroller, as shown in Figure 1.

The sensor's pressure measurements may be requested over the SPI bus. This protocol is also used for individual calibration of each device and so it is paramount that the communication protocol described in section 2.3 is implemented carefully. **Failure to do so may corrupt the device firmware.**

Figure 1



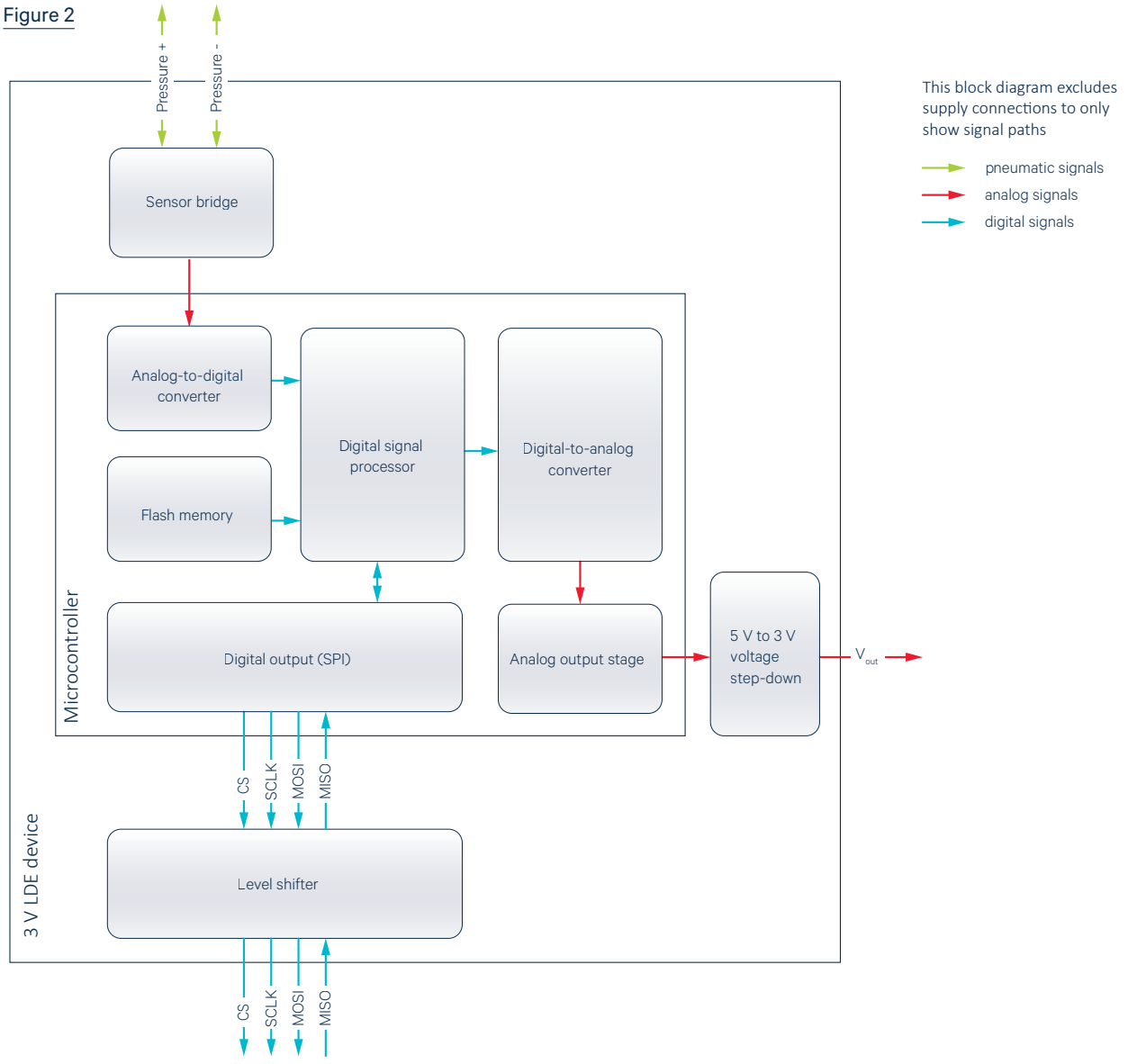
SPI bus communication with LDE/LME pressure sensors

2.2.2 3 V LDE devices

The 3 V LDE device is identical to the 5 V device, except for the use of a level shifter on the SPI bus, as shown in Figure 2. Additionally, but not shown, a voltage converter is used to provide the sensor bridge and the microcontroller with the required 5 V from the 3 V supply voltage.

This level shifter translates logic voltages from the native 5 V of the sensor microcontroller down to 3 V. Because of this hardware addition, the microcontroller driving the SPI bus must be capable of sourcing a minimum 2 mA current to ensure proper communication.

Figure 2



SPI bus communication with LDE/LME pressure sensors

2.3 Major specifications for SPI bus communication

The sensor's pressure measurement result may be requested over the SPI bus. As described in the LDE/LME datasheet, this is done by sending three successive commands, MSB first, to the sensor:

Step	Hexadecimal	Binary	Description
1	**0x2D**	**B00101101**	**Poll current pressure measurement**
2	**0x14**	**B00010100**	**Send result to data register**
3	**0x98**	**B10011000**	**Read data register**

Per the standard SPI protocol, each command is sent in the following manner:

1. Assert /CS low to enable communication with the sensor.
2. For each bit of the command (MSB first),
   - set the MOSI line to the correct logic level for the current bit;
   - pulse the SCLK line.
3. Assert /CS high to disable communication with the sensor.

The entire two-byte result is then read out on the MISO pin, MSB first, by applying 16 successive pulses on the SCLK line while /CS is asserted low.

/CS should be held high whenever the SPI master device is not communicating with the sensor.

It is also important to ensure that the timing between signals on the SPI bus adheres to the *Interface specification criteria* in the LDE/LME datasheet.

2.4 LDE evaluation kit

There is an evaluation kit available for LDE devices. Please contact First Sensor for details and purchasing information.

2.4.1 Difference between evaluation kit and datasheet communication protocols

The evaluation kit and its associated Windows® software interacts with the installed LDE devices using a proprietary communication protocol over the SPI bus.

This specialized protocol provides the evaluation software with the commands for requesting a pressure measurement as documented on the LDE datasheet, as well as access to otherwise non-user-accessible areas of the LDE device's flash memory.

Users should be mindful of this custom protocol should they attempt to observe the evaluation kit's SPI bus signals, as it is unlikely to aid any efforts in debugging their own code.

SPI bus communication with LDE/LME pressure sensors

3 Hardware connection considerations

For the sensor pinout please see the LDE/LME datasheet.

The connections listed under “Digital Output Connection” in the table below are required for digital communication over SPI bus.

Description	Digital Output Connection	LDE Pin	LME Pin
Reserved	**No connection**	**1**	**N/A**
Voltage supply	**Supply voltage**	**2**	**1**
Ground	**Ground**	**3**	**2**
Analog output	**No connection**	**4**	**3**
Analog output	**No connection**	**5**	**4**
SPI clock	**Master microcontroller’s SPI clock pin**	**6**	**5**
SPI Master Out Slave In (MOSI)	**Master microcontroller’s MOSI pin**	**7**	**6**
SPI Master In Slave Out (MISO)	**Master microcontroller’s MISO pin**	**8**	**7**
SPI slave select	**Master microcontroller’s slave select pin**	**9**	**8**
Reserved	**No connection**	**10**	**N/A**

3.1 Simple point-to-point connection

In the case of single LDE/LME 5 V devices (and LDE 3 V devices with a microcontroller capable of driving the SPI bus with 2 mA per line), the pins for the SPI bus may be directly connected between the sensor and the microcontroller. No further external components are necessary. The use of pull-up resistors is not recommended.

SPI bus communication with LDE/LME pressure sensors

3.2 Use of series resistors and optional buffers

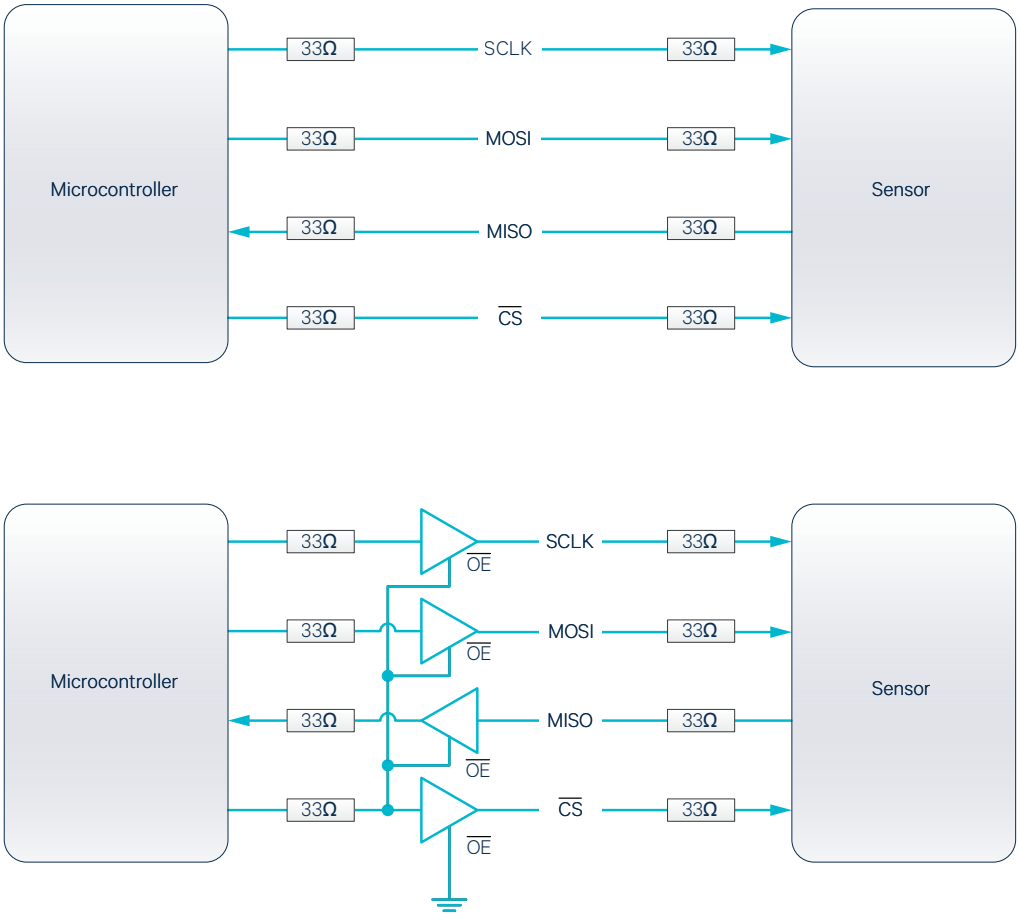
There are some application cases where 33 Ω series resistors at both ends of each SPI bus line may be helpful. Signal quality may be further improved by also adding buffers or line drivers, such as a 74HC125, on each line as well.

These application cases include:
- multiple slave devices on the same bus segment;
- using a master device with limited bus driving capability;
- long SPI bus lines.

Both of these topologies are shown in Figure 3.

If series resistors are used, they must be physically placed as close as possible to the pins of the master and slave devices.

Figure 3



SPI bus communication with LDE/LME pressure sensors

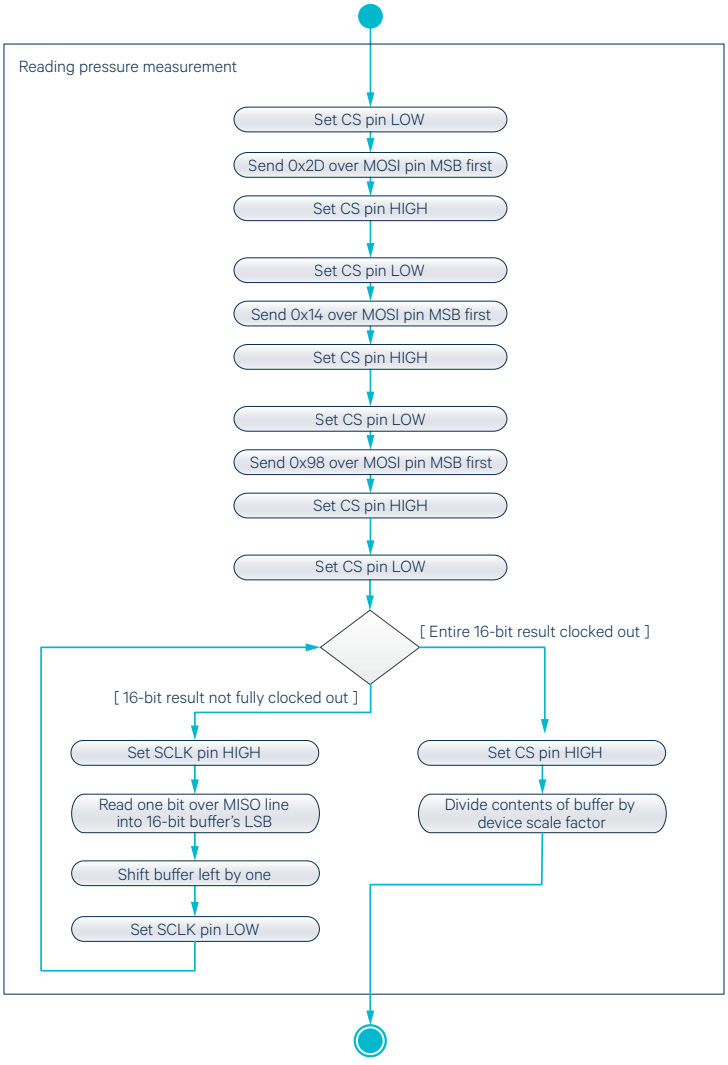
4 Software considerations

The flow of data to and from the LDE/LME device requires a very specific sequence of events that are controlled by software running on the SPI master device.

4.1 UML activity diagram

The UML activity diagram shown in Figure 4 describes the sequence of actions that must be performed by the SPI master device to read a pressure measurement in Pascals from an LDE/LME device.

Figure 4



SPI bus communication with LDE/LME pressure sensors

4.2 Sample code

The sample code below provides a very generic example of code that can be used to implement the pressure-reading operation described in Figure 4 above. The pin numbers indicated are arbitrarily chosen and will vary with the SPI master device being used.

Note that the microcontroller used also likely requires that the pins will be set as either inputs or outputs:

- CS_PIN → OUTPUT
- MOSI_PIN → OUTPUT
- SCLK_PIN → OUTPUT
- MISO_PIN → INPUT

This code listing ignores setting the pin mode as the method varies greatly between microcontrollers. This code listing further ignores the timing requirements for the bus for the same reason.

```
// Example pin number selected arbitrarily. This will vary
// depending on the microcontroller used.
const int CS_PIN = 10;
const int MOSI_PIN = 11;
const int MISO_PIN = 12;
const int SCLK_PIN = 13;

// Constants for commands to be sent to the LDE/LME device. Let the
// byte datatype used here be an array of either 0 or 1 values.
const uint8_t READ_LDE = 0x2D;
const uint8_t SEND2DHR = 0x14;
const uint8_t READ_DHR = 0x98;

// Set the scale factor for the LDE/LME device in question (see
// the LDE/LME datasheet for more information on scale factors). As an
// example, let's suppose we're using an LDES250UF6S device.
const int SCALE_FACTOR = 120;

// Initialize the SPI pins. We don't initialize the MISO pin as
// it is driven by the LDE/LME device.
CS_PIN = 1;
MOSI_PIN = 0;
SCLK_PIN = 0;

// Other constants used in the program.
const int dataDelay = 1;
const int clockDelay = 1;
```

SPI bus communication with LDE/LME pressure sensors

```
// The main program loop; gets a single pressure measurement.
// This can be looped continuously if necessary.
int main() {
    // Create a buffer to read the result into.
    uint16_t buffer;

    // Send the command to poll the current pressure reading.
    writeByte(READ_LDE);

    // Send the command to send the pressure reading to the
    // data register.
    writeByte(SEND2DHR);

    // Send the command to read the data register.
    writeByte(READ_DHR);

    // Read the first byte back from the LDE/LME sensor into
    // the low word of the buffer.
    buffer = readByte();

    // Shift this byte up into the high word of the buffer.
    buffer = buffer << 8;

    // Read the second byte back from the LDE/LME sensor into
    // the low word of the buffer.
    buffer |= readByte();

    // Get the value of the pressure measurement in Pascals.
    double result = buffer / SCALE_FACTOR;

    return 0;
}

// Send one byte to the sensor.
void writeByte(uint8_t toSend) {
    // Drive the CS pin low to enable communication with the slave.
    CS_PIN = 0;

    // Send the byte over the MOSI pin one bit at a time, MSB first.
    for (int i = 0; i < 8; i++) {
        // Drive the MOSI pin to the value of the current bit in the
        // byte to send.
        MOSI_PIN = (toSend & ( 1 << (7 - 1)));
        delayMicroseconds(dataDelay);
```

SPI bus communication with LDE/LME pressure sensors

```
        // Toggle the SCLK pin high.
        SCLK_PIN = 1;
        delayMicroseconds(clockDelay);

        // Toggle the SCLK pin low.
        SCLK_PIN = 0;
    }

    // Drive the MOSI pin low so that it's in a known state after the
    // byte has been sent.
    MOSI_PIN = 0;

    // Drive the CS pin high to disable communication with the slave.
    CS_PIN = 1;
}

// Read one byte from the sensor.
uint8_t readByte() {
    // Create a buffer to store the received byte.
    uint8_t data;

    // Drive the CS pin low to enable communication with the slave.
    CS_PIN = 0;

    // Get the byte from the LDE/LME device over the MISO pin, one bit at
    // a time, MSB first.
    for (int i = 0; i < 8; i++) {
        // Drive the SCLK pin high.
        SCLK_PIN = 1;

        // Read the value on the MISO pin.
        data |= !!MISO_PIN << (7 - i);
        delayMicroseconds(dataDelay);

        // Drive the SCLK pin low.
        delayMicroseconds(clockDelay);
        SCLK_PIN = 0;
    }

    // Drive the CS pin high to disable communication with the slave.
    CS_PIN = 1;
}
```

SPI bus communication with LDE/LME pressure sensors

5 Troubleshooting

To resolve difficulties in communicating with the sensor over SPI, the following steps may help:

1. If using an LDE device, confirm that the sensor works normally in a First Sensor evaluation kit to rule out firmware corruption or damage to the sensor.
2. Test code using a short, simple and direct point-to-point connection between the sensor and the SPI master device.
3. Ensure that the SPI bus lines are connected correctly:
   MISO ↔ MISO,
   MOSI ↔ MOSI,
   SCLK ↔ SCLK,
   /CS ↔ /CS
4. Ensure that the code implementation uses SPI Mode 0 (CPOL=0, CPHA=0).
5. Ensure that data is sent and received MSB first.
6. Visualize the SPI bus on an oscilloscope to confirm that commands are being sent correctly and that /CS is being toggled appropriately (HIGH on idle, LOW to enable).

Contact First Sensor for further information and troubleshooting help.