Digital Communication with SM5800 Series Parts

OVERVIEW

The SM5800 series pressure product offers the corrected pressure output in both analog and digital formats. Accessing the analog output is an easy process requiring a simple voltage measurement. But accessing the digital output requires an understanding of the digital communication protocols and memory locations containing the digital output data. The purpose of this note is to provide the information needed to establish a connection with a SM5800 part and then read the digital output of the part.

INTRODUCTION

The SM5800 series pressure product utilizes the Inter Integrated Circuit (I^2C) Bus interface for all digital communication. Using the I^2C bus the corrected digital pressure and digital temperature values are accessed from on board memory registers.

DIGITAL COMMUNICATION BASICS

The I^2C bus interface utilizes a two-wire method for carrying information back and forth between integrated circuit devices connected to the bus. The two-wire method is comprised of a bi-directional, 8-bit serial data (SDA) line providing the path for data transfers and a serial clock (SCL) line for synchronizing the data transfers.

A Master device in the form of a personal computer or microcontroller initiates and terminates data transfers on the SDA line and generates the clock signal on the SCL line. A Slave device (which represents the SM5800 series product) receives requests from the Master and transmits data according to the request. When multiple Slave devices are connected to the bus, each Slave must have a unique address and the Master must use this unique address when performing data transfers with a particular Slave. A diagram of the I^2C bus with a Master and multiple Slaves (or SM5800 parts) is shown in Figure 1.

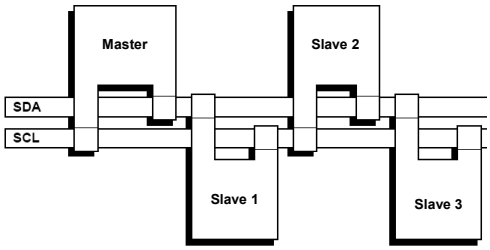


Figure 1. Multiple SM5800 series devices connected to the I^2C bus

DATA TRANSFER

The data transfer process begins with the Master issuing a START (S) command, which alerts all Slave devices of the pending transfer request. The Master then specifies which Slave device to communicate with and how the data will flow between the Master and Slave. Next, the Master waits for the addressed Slave to respond. When the desired Slave responds, the Master and Slave begin the process of transmitting data back and forth. The transfer process ends with the Master issuing a STOP (P) command. Figure 2 provides a diagram showing a simple data transfer sequence.

START & STOP CONDITIONS

Initiated by the Master, a START conditioned is identified by a HIGH to LOW transition of the SDA line while maintaining a stable HIGH condition on the SCL line. All data transfer must begin with a START condition and the bus lines are considered busy after a START condition is issued. To end the transfer of data the Master initiates a STOP condition, which is identified by a LOW to HIGH transition of the SDA line while maintaining a stable HIGH condition on the SCL line.

Silicon Microstructures, Inc. ♦ 1701 McCarthy Blvd. ♦ Milpitas, CA 95035 ♦ USA
Tel: 408-577-0100 ♦ Fax: 408-577-0123 ♦ Sales@si-micro.com ♦ www.si-micro.com

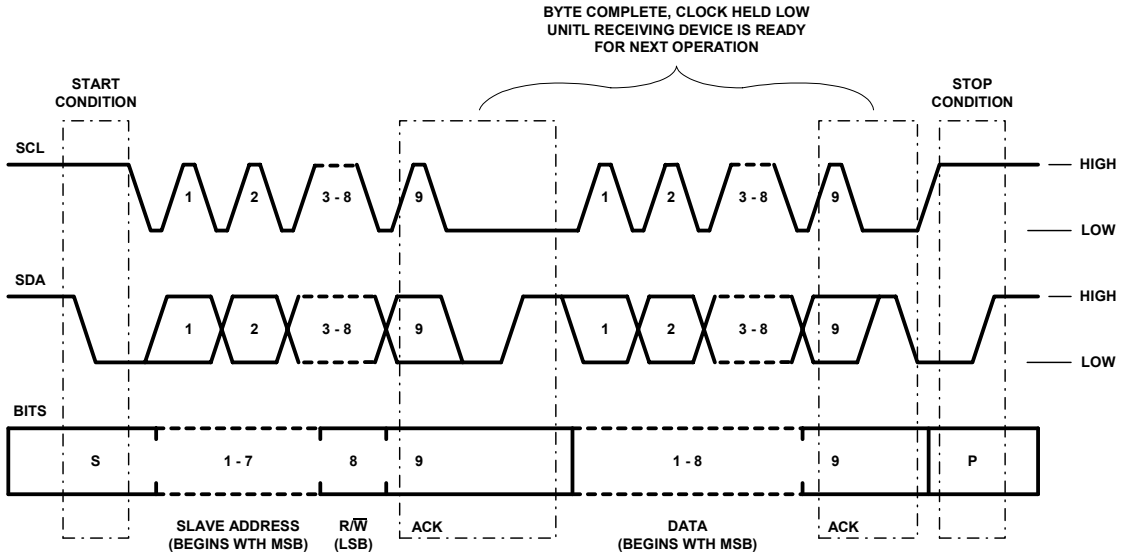


Figure 2. Diagram of a simple data transfer sequence

DATA FORMAT & VALIDITY

The SDA line is an 8-bit digital line and this defines the size of each byte transferred. An acknowledge bit starts with the Transmitter releasing the SDA line HIGH at the start of the clock cycle immediately following a byte transfer. The Receiver must then pull the SDA line LOW during the HIGH period of the same clock cycle. The set-up and hold times for the SDA and SCL lines are provided in APPENDIX A.

Every byte transferred from Transmitter to Receiver starts with the most significant bit (MSB). Valid data transfers only occur during the HIGH period of the SCL clock. If a Receiver needs time to carry out a request before receiving another byte, it can hold the SCL line LOW. To do this the Receiver performs an internal interrupt to hold the SCL LOW, which forces the Transmitter to wait before sending the next byte. When ready, the Receiver releases the SCL line and data transfer continues.

ADDRESSING A SLAVE DEVICE

All Slave devices must have a unique identification or address. A Slave address cannot exceed a length of seven bits. The Slave address shares space in the address byte with a read/write bit and the maximum length of the address byte is eight bits. To construct the address byte, place the Slave address at the MSB of the address byte and place the read/write bit at the least significant bit (LSB).

The address byte always follows a START condition. When an address byte is sent over the SDA line, each Slave compares the MSB of this byte with its own unique address. The Slave with the matching address becomes active and responds by sending an acknowledge bit along the SDA line to the Master. How the Slave responds to the next byte transmitted by the Master following the acknowledgement depends on the read/write bit. When the LSB of the address byte is “1” the Slave transmits data for the Master to read. When the LSB of the address byte is ”0” the Slave waits for the Master to write data to the Slave.

Silicon Microstructures, Inc. ♦ 1701 McCarthy Blvd. ♦ Milpitas, CA 95035 ♦ USA
Tel: 408-577-0100 ♦ Fax: 408-577-0123 ♦ Sales@si-micro.com ♦ www.si-micro.com

EEPROM SPECIFIC READ OPERATIONS

The Slave devices with onboard EEPROM contain an internal byte register, which will store the register address of the EEPROM register recently accessed by the Master. To set the value of the internal byte register a *Random Address Read* operation must be performed. Once the internal byte register is set then the value of the EEPROM register corresponding to the address found in the internal byte register can be read repeatedly through the use of the *Current Address Read* operation. The two read operations are detailed in the following:

Random Address Read

Following the initial START condition, the Master performs a dummy write operation to load the EEPROM register address into the internal byte register. The Slave acknowledges this dummy write operation. The Master then sends a Repeated START (Sr) condition. Following the repeated START condition, the Master again sends the Slave address byte with the read/write bit set to "1". The Slave acknowledges this and outputs the EEPROM register data corresponding to the address stored in the internal byte register. The Master does not acknowledge having received the byte output of the Slave, but instead terminates the data transfer session with a STOP condition. Pseudo code detailing the steps of the random address read operation can be found in APPENDIX B. Figure 3 shows a diagram of the random address read operation.

Current Address Read

Following a START condition, the Master transmits the first byte with the read/write set to "1". The Slave acknowledges this and outputs the EEPROM register data corresponding to the address stored in the internal byte register. The Master does not acknowledge having received the byte output of the Slave, but instead terminates the data transfer session with a STOP condition. Figure 4 shows a diagram of the current address read operation.

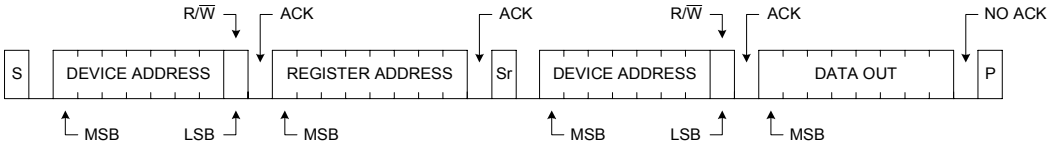


Figure 3. EEPROM specific Random Address Read operation

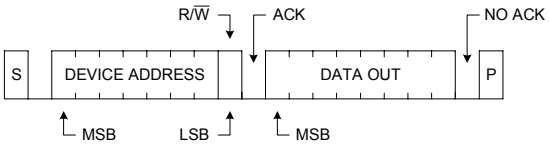


Figure 4. EEPROM specific Current Address Read operation

Silicon Microstructures, Inc. ♦ 1701 McCarthy Blvd. ♦ Milpitas, CA 95035 ♦ USA
Tel: 408-577-0100 ♦ Fax: 408-577-0123 ♦ Sales@si-micro.com ♦ www.si-micro.com

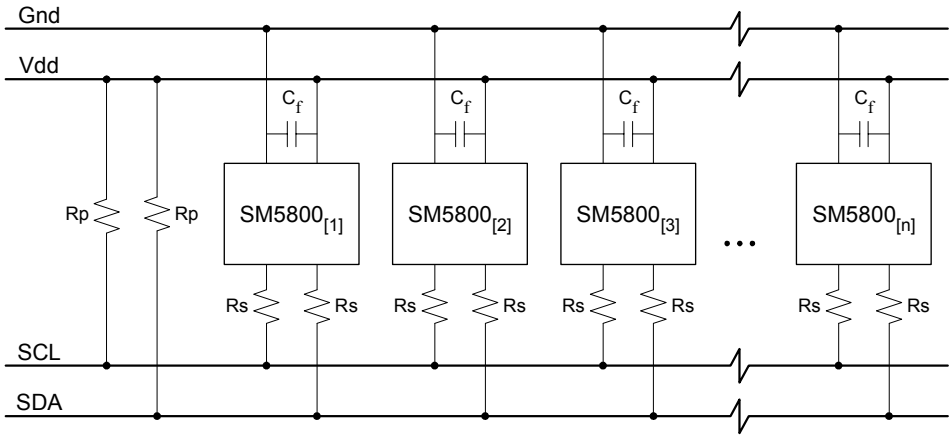


Figure 5. Implementation of Rp and Rs resistors along the I²C-bus

PRECAUTIONS

To protect devices connected to the SCL and SDA lines, series resistors must be utilized. The series resistors protect against high-voltage spikes traveling along the bus lines and help reduce interference and ringing. In addition to the series resistors, the SCL and SDA lines must be connected to the positive supply voltage (Vdd) using either a current-source or a pull-up resistor. A current-source or pull-up resistor helps maintain the HIGH state of the SCL and SDA lines and ensures that the signals are pulled from the LOW to HIGH state within the set rise time. For bus lines with a low capacitive load (<100pF) use pull-up resistors and when dealing with a high capacitive load (>100pF) use an external current-source pull-up. Figure 5 provides a diagram showing how to implement series and parallel resistors in the I^2C bus.

ACCESSING THE DIGITAL OUTPUT

A total of four digital values are accessible through the digital output of the SM5800. These digital values are:

- Band-gap voltage (11-bit)
- Temperature (11-bit)
- Uncorrected pressure (11-bit)
- Corrected pressure (12-bit)

The digital output values are stored in addressable memory registers and updated with each clock cycle. The length of the digital output values range from eleven to twelve bits. A single memory register has a maximum capacity of eight bits. Since a single digital output value exceeds the storage capacity of a single register a total of two consecutive memory registers are reserved for each digital output value. Each digital value is split in half with the LSB placed into the first register and the MSB placed into the second register. Table 1 provides a list of the register addresses reserved for the LSB and MSB of each digital output value.

Table 1. Digital Output Memory Mapping

REGISTER	DESCRIPTION
128	Corrected pressure, LSB [5:0]
129	Corrected pressure, MSB [11:6]
130	Temperature, LSB [5:0]
131	Temperature, MSB [11:6]
132	Un-corrected pressure, LSB [5:0]
133	Un-corrected pressure, MSB [11:6]
134	Band-gap voltage, LSB [5:0]
135	Band-gap voltage, MSB [11:6]

Silicon Microstructures, Inc. ♦ 1701 McCarthy Blvd. ♦ Milpitas, CA 95035 ♦ USA
Tel: 408-577-0100 ♦ Fax: 408-577-0123 ♦ Sales@si-micro.com ♦ www.si-micro.com

CONCLUSIONS

SM5800 series parts are now capable of digital communication through an I^2C bus interface. The two-wire I^2C bus allows multiple SM5800 series parts to be accessed. Different SM5800 models and pressure ranges can be connected to the same I^2C bus as long as each SM5800 series part has a unique bus address. Please consult the factory to obtain SM5800 series parts programmed with addresses other than the factory default value of *95*.

The digital interface provides access to the digital band-gap voltage, digital temperature, uncorrected digital pressure, and corrected digital pressure values of a SM5800 series part. Now a SM5800 series part can be used in a purely digital system thus eliminating the need of an analog-to-digital converter for the corrected analog pressure output.

APPENDIX A. *I^2C Bus Characteristics*

PARAMETER	MIN	TYP	MAX	UNITS
SCL frequency	0	–	100	kHz
LOW period of the SCL clock	4.7	–	–	µs
HIGH period of the SCL clock	4.0	–	–	µs
Rise time for SCL and SDA bus signals	–	–	1000	ns
Fall time for SCL and SDA bus signals	–	–	300	ns
Set-up time for repeated START condition	4.7	–	–	µs
Set-up time for Data	250	–	–	ns
Set-up time for STOP condition	4.0	–	–	µs
SDA transfer rate	–	–	100	kbits/s
LOW level input voltage for each bus line	-0.5	–	0.3 * Vdd	V
HIGH level input voltage for each bus line	0.7 * Vdd	–	–	V
Supply Voltage, Vdd	–	5	–	V
Supply Voltage Filter Capacitance, Cf		0.1		pF
Capacitive load per bus line, Cb	–	–	400	pF
Parallel Resistor, Rp	–	10	–	kΩ
Series Resistor, Rs	–	100	–	Ω
Noise limit at the LOW level for each device connected	0.1 * Vdd	–	–	V
Noise limit at the HIGH level for each device connected	0.2 * Vdd	–	–	V

Silicon Microstructures, Inc. ♦ 1701 McCarthy Blvd. ♦ Milpitas, CA 95035 ♦ USA
Tel: 408-577-0100 ♦ Fax: 408-577-0123 ♦ Sales@si-micro.com ♦ www.si-micro.com

APPENDIX B. *Random Address Read* Pseudo Code

```
//********** MAIN PROGRAM **********
START
  START_CONDITION;
  DEVICE_ADDRESS;                          //Set the read/write bit to "0" (= write)
  ACKNOWLEDGEMENT;
  REGISER_ADDRESS;
  ACKNOWLEDGEMENT;
  RE-START_CONDITION:
  DEVICE_ADDRESS;                          //Set the read/write bit to "1" (= read)
  ACKNOWLEDGEMENT;
  DATA_OUT;
  NO_ACKNOWLEDGEMENT;
  STOP_CONDITION;
END

//********** SUB-ROUTINES **********
START_CONDITION
  SDA(W)=0;                                //Set SDA bus line "low"
  DELAY;
  SCL(W)=1;                                //Set SCL bus line "high"
  DELAY;

DEVICE_ADDRESS
  n=7;
  FOR i=0 to n-1
    SDA(W)=DEVICE_ADD(i);                  //Writing ith bit of the Device Address
    DELAY;
    SCL(W)=0;                              //Set SCL bus line "low"
    DELAY;
    SCL(W)=1;                              //Set SCL bus line "high"
    DELAY
    i=i+1;
  END;
  SDA(W)=R_or_W;                           //Set the read/write bit
  DELAY;
  SCL(W)=0;                                //Set SCL bus line "low"
  DELAY;
  SCL(W)=1;                                //Set SCL bus line "high"
  DELAY;

ACKNOWLEDGEMENT
  SDA(W)=1;                                //Set SDA bus line "high"
  DELAY;
  SCL(W)=0;                                //Set SCL bus line "low"
  DELAY;
  ACK=SDA(R);                              //Reading ACK bit
  DELAY;
  SCL(W)=1;                                //Set SCL bus line "high"
  DELAY;

REGISTER_ADDRESS
  n=8;
  FOR i=0 to n-1
    SDA(W)=REGISTER_ADD(i);                //Writing ith bit of the Register Address
    DELAY;
    SCL(W)=0;                              //Set SCL bus line "low"
    DELAY;
    SCL(W)=1;                              //Set SCL bus line "high"
    DELAY
    i=i+1;
  END;
```


Silicon Microstructures, Inc. ♦ 1701 McCarthy Blvd. ♦ Milpitas, CA 95035 ♦ USA
Tel: 408-577-0100 ♦ Fax: 408-577-0123 ♦ Sales@si-micro.com ♦ www.si-micro.com

```
RE-START_CONDITION
  SDA(W)=1;                              //Set SDA bus line "high"
  DELAY;
  SCL(W)=0;                              //Set SCL bus line "low"
  DELAY;
  SDA(W)=0;                              //Set SDA bus line "low"
  DELAY;
  SCL(W)=1;                              //Set SCL bus line "high"
  DELAY;

DATA_OUT
  n=8;
  FOR i=0 to n-1
    DATA(i)=SDA(R);                      //Reading ith bit of Data Out
    DELAY;
    SCL(W)=0;                            //Set SCL bus line "low"
    DELAY;
    SCL(W)=1;                            //Set SCL bus line "high"
    DELAY
    i=i+1;
  END;

NO_ACKNOWLEDGEMENT
  SDA(W)=1;                              //Set SDA bus line "high"
  DELAY;
  SCL(W)=0;                              //Set SCL bus line "low"
  DELAY;
  SCL(W)=1;                              //Set SCL bus line "high"
  DELAY;

STOP_CONDITION
  SDA(W)=0;                              //Set SDA bus line "low"
  DELAY;
  SCL(W)=0;                              //Set SCL bus line "low"
  DELAY;
  SDA(W)=1;                              //Set SDA bus line "high"
  DELAY;
```

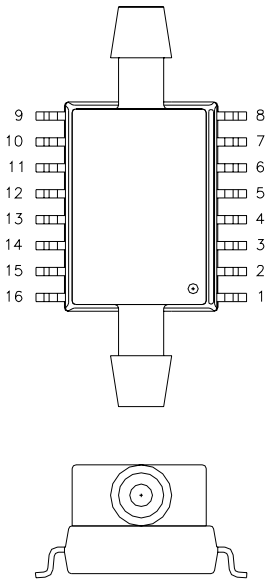

Silicon Microstructures, Inc. ♦ 1701 McCarthy Blvd. ♦ Milpitas, CA 95035 ♦ USA
Tel: 408-577-0100 ♦ Fax: 408-577-0123 ♦ Sales@si-micro.com ♦ www.si-micro.com

APPENDIX C. Glossary of Terms

ACKNOWLEDGE (ACK) BIT :	LOW signal sent by the over the SDA line during the 9th clock pulse indicating a successful data transfer.
ADDRESS :	7-bits used to identify a specific SLAVE on the I^{2}C-bus and found in the MSB of the first byte transmitted after a START condition.
BUS CAPACITENCE (Cb) :	Total capacitance of the wire, connections, and pins found on the bus.
I^{2}C Bus :	Inter Integrated Circuit, a two-wire communication bus consisting of a bi-directional serial data line and a clock line. (Also known as *Inter IC*.)
MASTER :	A desktop computer or microcontroller. Generates the clock signal and initiates/terminates the data transfer process with a Slave.
PARALLEL RESISTOR (Rp) :	Resistor placed between the bus and supply voltage lines.
READ (R) :	8th bit of the ADDRESS byte with a value of "1".
RECEIVER :	Obtains data transmitted over the SDA line.
REPEATED START CONDITION (Sr) :	See START Condition. Occurs during the data transfer process.
SERIAL CLOCK (SCL) :	Line used to synchronize data transfers between a Master and Slave device.
SERIAL DATA (SDA) :	Bi-directional line used to transfer data between a Master and Slave device.
SERIES RESISTOR (Rs) :	Resistor placed between the bus line and a Slave device.
START CONDITION (S) :	Transitioning from HIGH to LOW on the SDA line while the SCL line remains HIGH.
STOP CONDITION (P) :	Transitioning from LOW to HIGH on the SDA line while the SCL line remains HIGH.
SUPPLY VOLTAGE (Vdd) :	Voltage powering I^{2}C devices.
SLAVE :	A SM5800 series part. Receives/sends data when addressed by a Master.
TRANSMITTER :	Sends data over the SDA line.
$WRITE(\overline{W})$:	The 8th bit of the ADDRESS byte with a value of "0".

Silicon Microstructures, Inc. ♦ 1701 McCarthy Blvd. ♦ Milpitas, CA 95035 ♦ USA
Tel: 408-577-0100 ♦ Fax: 408-577-0123 ♦ Sales@si-micro.com ♦ www.si-micro.com

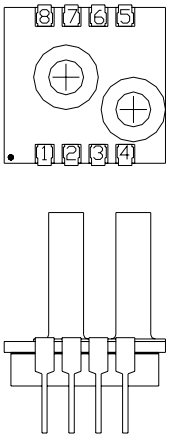
APPENDIX D. SM5800 Package Pin-Out Diagrams



PIN NO.	PIN DESCRIPTION
1	NC
2	NC
3	SERIAL DATA (SDA)
4	SERIAL CLOCK (SCL)
5	NC
6	NC
7	NC
8	NC
9	NC
10	NC
11	NC
12	GROUND
13	Vexc = 5.000 VDC
14	NC
15	NC
16	ANALOG OUTPUT

NOTE: DO NOT GROUND NC PINS.
NC PINS MUST FLOAT.

Figure 6. SM5822/SM5872 package pin-out diagram



PIN NO.	PIN DESCRIPTION
1	NC
2	GROUND
3	NC
4	SERIAL DATA (SDA)
5	SERIAL CLOCK (SCL)
6	NC
7	Vexc = 5.000 VDC
8	ANALOG OUTPUT

NOTE: DO NOT GROUND NC PINS.
NC PINS MUST FLOAT.

Figure 7. SM5812/SM5852 package pin-out diagram

Silicon Microstructures, Inc. ♦ 1701 McCarthy Blvd. ♦ Milpitas, CA 95035 ♦ USA
Tel: 408-577-0100 ♦ Fax: 408-577-0123 ♦ Sales@si-micro.com ♦ www.si-micro.com