

Application Note

1 General product description

The integrated circuit MS1089A is a fully integrated calibrated digital low power temperature sensor with a typical temperature measurement accuracy of $\pm 0.3^{\circ}\text{C}$. The MS1089A has an I²C interface and is available in Chip-Scale-Package (CSP).

The Datasheet and further information is available on the [Microdul Home Page](#).

2 Operating the MS1089A

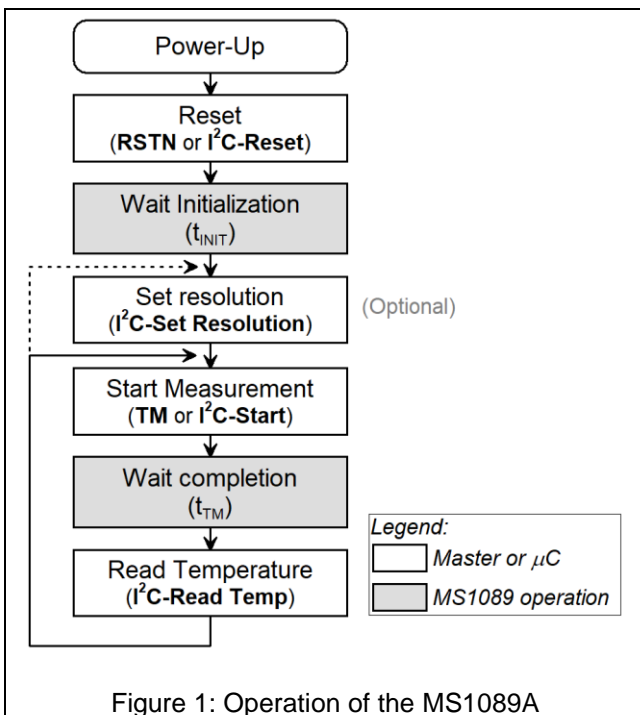


Figure 1: Operation of the MS1089A

Figure 1 illustrates the operation flow of the MS1089A.

3 Initialisation after power-up

After power-up the MS1089A must be initialized with a Reset. A Reset can either be done by setting input RSTN LOW or by software with the I²C Reset command. It is advised to use the RSTN pin after power-up to correctly initialize the MS1089A.

If not initialized with a Reset, the thermometer of the MS1089A is not calibrated and its accuracy is not guaranteed. After power-up and until a Reset is applied, the current consumption is undefined and may reach approximately 150 μA in the worst case.

The duration of the initialization phase t_{INIT} after the reset takes maximum 2.1 ms. It is advisable not to issue any I²C command or TM pulse during the initialization phase.

3.1 Initialisation with RSTN pin

A typical reset sequence using the RSTN pin is shown in the figure below. The RSTN low-level pulse must be applied when the V_{DD} supply voltage has reached at least 1.6 V. Alternatively the RSTN pin can also be held at low level, until the V_{DD} supply voltage has reached at least 1.6 V.

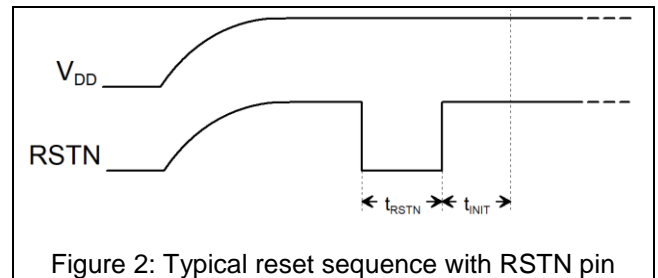


Figure 2: Typical reset sequence with RSTN pin

The RSTN pulse width must have a duration of at least 20 ns. The figure also illustrates the waiting time t_{INIT} of at least 2.1 ms which is required for proper initialization of the MS1089A.

3.2 Initialisation with I²C command

If the pin RSTN cannot be used to initialize the MS1089 after power-up, the I²C command I4 'Chip reset' shall be used for that purpose. To ensure that the I²C interface itself is in a defined state, the chip reset command I4 should be preceded by a general call (I²C command to address 00h) with one dummy data byte. This sequence is illustrated in Figure 3.

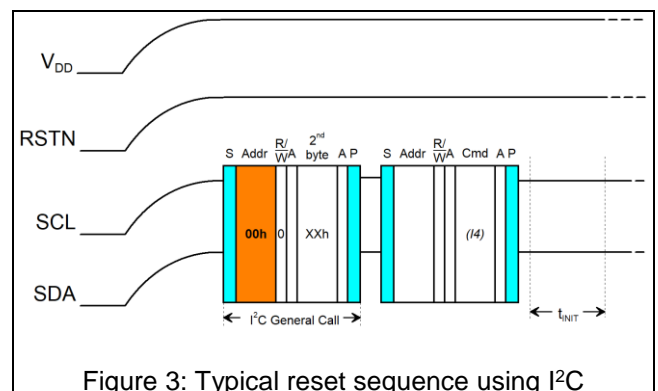


Figure 3: Typical reset sequence using I²C

The first I²C interaction should not take place before the V_{DD} supply voltage has reached at least 1.6 V. The figure also illustrates the waiting time t_{INIT} of at least 2.1 ms after the I4 command, which is required for the proper initialization of the MS1089A.

4 Temperature measurement

4.1 Resolution

The MS1089A offers 3 selectable temperature measurement resolutions: 0.1°C (11 bit), 0.05°C (12 bit) and 0.025°C (13 bit). After reset, the temperature measurement resolution of the MS1089A is set to 0.1°C (11 bit). This resolution leads to the shortest measurement duration, and therefore to the lowest power consumption. Higher resolution settings come at the cost of increased measurement duration and power consumption.

A typical temperature measurement duration t_M at room temperature and 11 bit resolution is 24 ms, and at 13 bit resolution the measurement takes 94 ms.

4.2 Measurement sequence

The temperature measurement can be started with the I²C command I2 or with a low-level pulse on the hardware handshake pin TM. After completing the measurement, the MS1089A itself pulls line TM low in both cases.

Using the bidirectional TM handshake mechanism enables the lowest power consumption, because the TM start pulse consumes less power than the I²C

command I2, and the waiting time for the measurement result can be optimized.

The timing requirements timing diagrams of the two measurement sequences are described in more detail in the MS1089A datasheet.

4.3 Temperature data format

After completion of the measurement, the temperature data must be read from the MS1089A with the I1 command. The resulting temperature data value TD is an unsigned 16-bit integer. The actual temperature value is calculated as follows from TD:

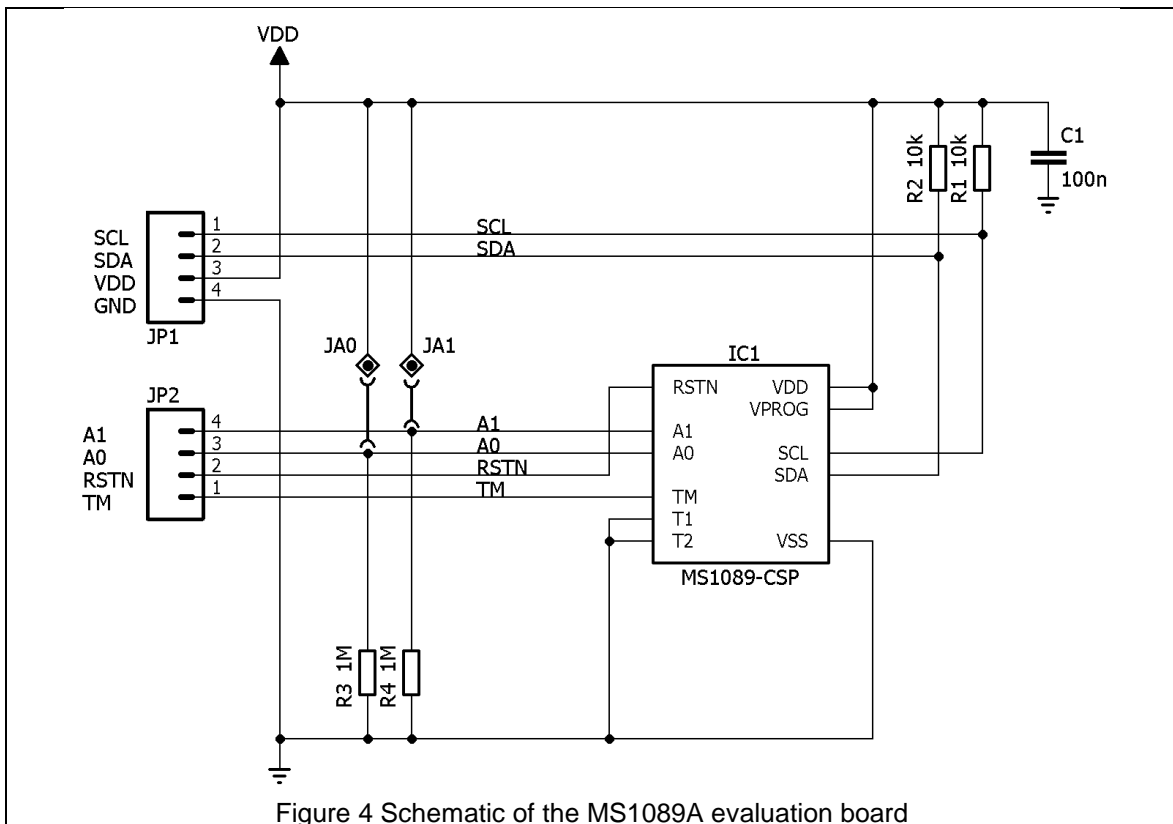
$$T(^{\circ}C) = \frac{TD}{40} - 80$$

$$T(^{\circ}F) = \left(\frac{TD}{40} - 80\right) \times 1.8 + 32$$

4.4 Temperature error at VDD < 2.2V

At supply voltages VDD < 2.2 V the temperature measurement error of the MS1089A increases moderately. Please refer to the datasheet for detailed figures.

5 Evaluation board



The schematic of the evaluation board is shown in Figure 4. The MS1089A signals are available on the connectors JP1 and JP2. The pinning of the

connectors is given in Table 1. The jumpers JA0 and JA1 allow an easy on-board selection of the four available I²C addresses of the MS1089A.

Table 1: Connectors on the evaluation board

Pin	JP1	JP2
1	SCL	TM
2	SDA	RSTN
3	VDD	A0
4	GND (VSS)	A1

Figure 5 shows the top side of the assembled evaluation board. The two connectors JP1 and JP2 are located on the lower side, and pin 1 of connectors JP1, JP2 are marked with a white dot (●) located next to it.

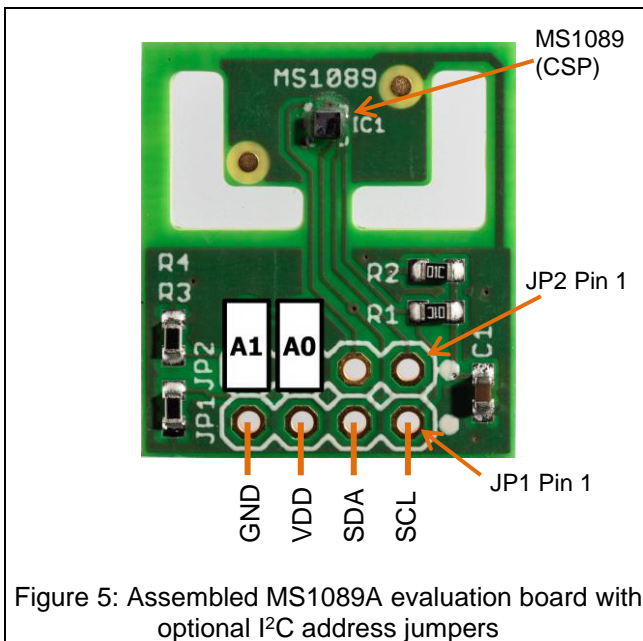


Figure 5: Assembled MS1089A evaluation board with optional I²C address jumpers

The figure also illustrates the jumper positions to set address pins A0 and/or A1 to high level. The evaluation board allows testing all functions of the MS1089A temperature sensor. The cutouts of the PCB thermally isolate the sensor IC from the component section and the connectors. This allows for a faster adaptation of the sensor to a changing ambient temperature.

5.1 I²C bus speed

The evaluation board contains two pull-up resistors R1 and R2 of 10 kΩ each, attached to the I²C bus lines SCL and SDA. The MS1089A supports the Fm+ (fast mode plus) bus speed of max. 1 MHz. To achieve this bus speed, the value of the on-board pull-up resistors may be too high. If this is the case, the resistors can be replaced with smaller values, or resistors with smaller values can be added in parallel to R1 and R2.

6 Coding requirements & examples

6.1 Coding requirements

As mentioned above and described in detail in the datasheet of the MS1089A, the following precautions must be taken in the application code:

- If no hardware reset is possible, a software reset over I²C is mandatory.
- After the reset, the waiting time t_{INIT} must be obeyed before further interaction with the MS1089A is recommended.
- If the hardware handshake line TM is not used to detect the end of the measurement, a software delay to wait for the completion of the measurement is required.

6.2 Arduino code example

The following code is an example which runs on the Arduino platform. It initialises the MS1089A using the I²C interface and then periodically measures temperature every 2 seconds. Both, the digital temperature value as well as the real value in °C, are printed to the serial terminal.

The example source code can be downloaded from the Microdul web page:

<https://www.microdul.com/en/ultra-low-power-sensors/temperaturesensor/>

I²C address and other definitions:

```
// I2C address definition
#define I2C_ADDR 0x44 // A0 = A1 = GND
// #define I2C_ADDR 0x45 // A0 = VDD & A1 = GND
// #define I2C_ADDR 0x46 // A0 = GND & A1 = VDD
// #define I2C_ADDR 0x47 // A0 = A1 = VDD

// Command codes
#define I1_RDTMP 0x00
#define I2_START 0x01
#define I3_RESOL 0x02
#define I4_RESET 0x03

// Resolution settings
#define RES_11B 0x00
#define RES_12B 0x01
#define RES_13B 0x02
```

Initialisation of the MS1089A and configuration for 12-bit temperature measurement is done in the `setup()` routine:

```
void setup() {
  // Setup communications, I2C 400 kHz mode
  Serial.begin(115200);
  Wire.begin();
  Wire.setClock(400000);

  // Turn off Arduino onboard LED
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW);

  // MS1089 reset sequence
  write_command(0x00, 0x00); // General call
  write_command(I2C_ADDR, I4_RESET); // Reset command
  delay(5); // Wait for Init

  // Set 12-bit resolution
  write_data(I2C_ADDR, I3_RESOL, RES_12B);
}
```

In the main loop, the Arduino flashes a LED, then performs a temperature measurement and prints the result to the serial terminal. A delay at the end of the main loop makes sure the measurement is performed approximately every 2s:

```
void loop() {
  uint16_t TD;
  float tempC;

  // Flash LED before measurement
  digitalWrite(LED_BUILTIN, HIGH);
  delay(100);
  digitalWrite(LED_BUILTIN, LOW);

  // Start temperature measurement
  write_command(I2C_ADDR, I2_START);
  // Wait for completion (safe value for all resolutions)
  delay(150);
  // Read the digital temperature data value
  TD = read_data_2b(I2C_ADDR, I1_RDTMP);

  // Calculate the real value in °C
  tempC = ((float)TD / 40.0) - 80.0;

  // Data output to Serial monitor
  Serial.print(TD);
  Serial.print(", ");
  Serial.println(tempC);

  // Loop delay to get a measurement ~ every 2 s
  delay(1750);
}
```

The following subroutines simplify the I²C access to the MS1089A:

```
/* I2C access routines
******/

void write_data(uint8_t address, uint8_t command, uint8_t databyte)
{
  Wire.beginTransmission(address);
  Wire.write(command);
  Wire.write(databyte);
  Wire.endTransmission();
}

uint16_t read_data_2b(uint8_t address, uint8_t command)
{
  uint16_t value = 0;

  Wire.beginTransmission(address);
  Wire.write(command);
  Wire.endTransmission(false);
  Wire.requestFrom(address, (uint8_t) 2);
  value = Wire.read();
  value = (value << 8) + Wire.read();
  return (value);
}

void write_command(uint8_t address, uint8_t command)
{
  Wire.beginTransmission(address);
  Wire.write(command);
  Wire.endTransmission(true);
}
```

7 ESD

Inputs and outputs are protected against electrostatic discharge during normal operation. However to be totally safe, it is advisable to undertake precautions appropriate to handling MOS devices in all process steps.

8 Disclaimer

Whilst every effort is taken to make sure that the information contained in this document is correct, Microdul AG accepts no liability whatsoever for the accuracy or completeness of the information given. Microdul AG reserves the right to change or correct information without prior notice as necessary.

9 Ordering information

For ordering MS1089A samples and the MS1089A evaluation board, please contact:

semiconductors@microdul.com