TSYS03
Digital Temperature Sensor

Application Note

Programming an alternative I2C-address
The TSYS03 temperature sensor offers the option to write an alternative I2C-address statically to the memory that will be used by the TSYS03 to communicate.

If this is done post production, the CRC value of the von-volatile memory (NVM) will be wrong after the new I2C address is written. However the sensor will work fine and perform normally.

The original and alternative I2C-Address will behave the same. They are both written to the 0x40 address by default. The user can program a different address to the NVM.

The alternative address can be changed between 0x40 to 0x7E (64dec to 126dec). Please note, ONLY the even addresses can be used.

The LSB of the 7bit address is always set to 0.

**Changing the address of TSYS03:**

**ATTENTION:** Writing to the NVM could cause irreparable damage to the sensor !!

- Connect TSYS03
- Send a „Service-Unlock”-command
- Write address to the NVM
- Read address from NVM to confirm
- Reset the TSYS03

**NVM-Memory**

The NVM is structured and written word-wise. The alternative I2C-Address can be found in Configregister2 (address 5). The LSB is not mapped. Due to that, only the bits 1 to 6 of the address are written to the memory and these are stored in bits 8 to 13 of Configregister.

<table>
<thead>
<tr>
<th>Address</th>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>res</td>
<td>res</td>
<td>Alternative 2C (6:1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Service Command**

<table>
<thead>
<tr>
<th>Name</th>
<th>Reset</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service Unlock</td>
<td>0000</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>NVM write</td>
<td>0111</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>adr3</td>
<td>adr2</td>
<td>adr1</td>
<td>adr0</td>
<td>0</td>
</tr>
<tr>
<td>NVM read</td>
<td>1100</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>adr3</td>
<td>adr2</td>
<td>adr1</td>
<td>adr0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example for an alternative address: 0x46 (70dec)**

0x46 = 0b01000110 LSB is set by external circuit. The address has only 7 bit, so there is no 8 bit. Shift the address by 1 to the right → 0b00100011

The rest of the word is written as 0. So, a possible existing value isn’t overwritten.

The value, written to the config register is: 0x2300 (0b 0010 0011 0000 0000).

**Code example:**

```c
iByte[0] = 0x00
iByte[1] = ((iI2cAdressAlt >> 1)&0x3F); //right shift by 1 and mask
ConfigWord = iByte[1]<<8 | iByte[0] //build the word to write into configregister
```
I2C commands

Service Unlock

```
+-------------------------------+
| 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 |
+-------------------------------+
                  S Device Address | W A | cmd byte | A P |
```

From Master  S = Start Condition  W = Write  A = Acknowledge
From Slave  P = Stop Condition  R = Read  N = Not Acknowledgement

NVM Write

```
+-------------------------------+
| 1 0 0 0 0 0 0 0 0 0 0 0 1 0 1 | ADDR# |
+-------------------------------+
                  S Device Address | W A | cmd byte | A NVM data bit 15 - 8 A NVM data bit 7 - 0 |
```

From Master  S = Start Condition  W = Write  A = Acknowledge
From Slave  P = Stop Condition  R = Read  N = Not Acknowledgement

Period of dummy bits following data bits is to burn the memory

```
x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x 0
```

NVM Read

```
+-------------------------------+
| 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 | ADDR# |
+-------------------------------+
                  S Device Address | W A | cmd byte | A P S Device Address | R A dummy bit 31 - 24 A dummy bit 23 - 16 |
```

From Master  S = Start Condition  W = Write  A = Acknowledge
From Slave  P = Stop Condition  R = Read  N = Not Acknowledgement

Period of dummy bits is to load NVM data into register and then send to SDA for read

```
x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x 0
```

Reset

```
+-------------------------------+
| 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 |
+-------------------------------+
                  S Device Address | W A | cmd byte | A P |
```

SENSOR SOLUTIONS // APPLICATION NOTE TSYS03 06/2021