Embedded Systems

Praktikum 9

Fachhochschule Bielefeld
Campus Minden
Studiengang Informatik

Beteiligte Personen:

Name
Hendrik Schröder
Diana Kirzhner

Aufgaben:

Aufgabe	Gelöst
Aufgabe 1	Komplett
Aufgabe 2	Komplett

---

1. Juli 2018

Inhaltsverzeichnis

1 Aufgabe 1

1.1 a)

Wie lautet die Adresse des DS1621, mit dem dieser in der dargestellten Schaltung angesteuert wird? Erklären Sie, wie sich die Adresse zusammensetzt und wie Sie die Adresse mit der Wire-Bibliothek benutzen.

Die Adresse in der dargestellten Schaltung des DS1621 lautet 0x90 ¿¿ 1.
Durch die 0x90 erfolgt die Darstellung der Adresse in Hexadezimal durch Umwandlung in das Binäre-System sieht diese wie folgt aus 1001 0000 und bildet eine 8-Bit Adresse die vom I2C-Bus verwendet werden kann.
Durch das letzte Bit des Kontrollbytes in einer 8-Bit-Adresse wird die Auszuführende Operation definiert. Beim setzen einer "1ërfolgt die Auswahl einer Leseoperation bei einer "0ëiner Schreiboperation.

Da die Wire-Bibliothek nur durchgehend 7-Bit-Adressen verwendet ist es notwendig zu shiften, wodurch das Low-Bit gelöscht wird (dh der Wert um ein Bit nach rechts verschoben wird) und man somit eine 7-Bit-Adresse zwischen 0 und 127 erhält.
Zum Ansprechen des DS1621 im Code sowie im weiteren Programmverlauf verwenden wir eine 7-Bit-Adresse (Binäre Darstellung: 100 1000).
Das Kontrollbyte besteht aus einem 4-Bit-Kontrollcode. Für den DS1621 wird dieser als 1001-Binär für Lese- und Schreiboperationen festgelegt.
Bei den nachfolgenden 3 Bits des Steuerbytes handelt es sich um die Geräteauswahlbits (A2, A1, A0) =¿ effektiv die 3 niedrigstwertigen Bits der Slave-Adresse.
Diese bestimmen welches Gerät angesprochen werden soll, da man insgesamt 8 DS1621 Geräte anschließen kann.
In unserer Schema sind alle 3 Pins (A2, A1, A0) von DS1621 mit dem "GROUND"verbunden, was bedeutet, dass diese 3 Bits die Werte "000"haben.

1.2 b)

Wie setzt sich das Format zur Übertragung von Temperaturwerten bei dem DS1621 zusammen?
Das digitale Thermometer und Thermostat DS1621 liefert 9-Bit-Temperaturmesswerte, die die Temperatur des Geräts anzeigen. Die

9-Bit Temperaturwerte werden durch alle MSB-Bits und den ersten Bit des LSB dargestellt.

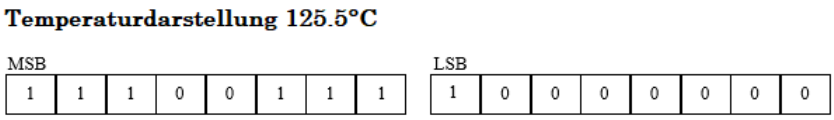

Abbildung 1: Beispiel für 125.5 °C Darstellung

Temperaturdaten vom DS1621 können entweder als einzelnes Byte (Temperaturauflösung von 1 °C) oder als zwei Bytes geschrieben /gelesen werden.
Das zweite Byte würde den Wert des niedrigstwertigen (0.5 °C) Bits des Temperaturmesswerts enthalten.
Die verbleibenden 7 Bits des LSB werden alle auf "0"gesetzt.

Welche Befehle und Parameter (mit der Wire-Bibliothek) müssen an den DS1621 gesendet werden, um dessen Pin 3 (TOUT) so einzustellen, dass dieser bei einer Temperatur größer 25 °C *auf HIGH geht?*

Code:

```
1  void setup()
2  {
3      Serial.begin(9600);
4      Wire.setModule(0);
5
6      Wire.begin();
7      Wire.beginTransmission(DEV_ID);              // connect to
           DS1621 (#0)
8      Wire.write(0xAC);                            // Access Config
9      Wire.write(0x2);                             // set for
           continuous conversion
10     Wire.endTransmission();
11     Wire.beginTransmission(DEV_ID);              // restart
12     Wire.write(0xEE);                            // start
           conversions
13     Wire.endTransmission();
14
15     // Tout ist aktiv wenn TH = 25.0° überschritten wird und
           setzt sich zurück, sobald der Wert unter TL= 24.0° fällt
16
17     Wire.beginTransmission(DEV_ID);              // restart
18     Wire.write(0xA1); // change TH
```

```
19      Wire.write(0x19); // Wert 25
20      Wire.write(0x0);  // Wert 0
21      Wire.endTransmission();
22
23      Wire.beginTransmission(DEV_ID);             // restart
24      Wire.write(0xA2); // change TL
25      Wire.write(0x18); // Wert 24
26      Wire.write(0x0);  // Wert 0
27      Wire.endTransmission();
28  }
```

1.3 c)

Mit welchen Anweisungen bekommen Sie die Temperatur vom DS1621 geliefert?

Code:

```
1   void setup()
2   {
3       Serial.begin(9600);
4       Wire.setModule(0);
5
6       Wire.begin();
7       Wire.beginTransmission(DEV_ID);             // connect to
            DS1621 (#0)
8       Wire.write(0xAC);                           // Access Config
9       Wire.write(0x2);                            // set for
            continuous conversion
10      Wire.endTransmission();
11      Wire.beginTransmission(DEV_ID);             // restart
12      Wire.write(0xEE);                           // start
            conversions
13      Wire.endTransmission();
14          (...)
15  }
16
17  void loop()
18  {
19      int8_t firstByte = 0;
20      int8_t secondByte = 0;
21      float result = 0;
22
23      delay(3000);// give time (3 sec) for measurement
24
25      //Anweisungen die die Temperatur liefern
```

```
26      Wire.beginTransmission(DEV_ID);
27      //Dieser Befehl liest das letzte
            Temperaturumwandlungsergebnis.
28     //Der DS1621 sendet 2 Bytes in dem zuvor beschriebenen
           Format, die den Inhalt dieses Registers darstellen.
29      Wire.write(0xAA);
30
31      Wire.endTransmission();
32
33          //Anweisungen werden angefordert
34      Wire.requestFrom(DEV_ID, 2);// ATTENTION: request from
            DS1621 (#0) two bytes (0.5 deg. resolution)
35
36          //Anweisungen (wenn vorhanden) werden gelesen
37      if(Wire.available())
38      {
39        firstByte = Wire.read();// get first byte!
40        Serial.print("firstByte");
41        Serial.println(firstByte);
42      }
43      else
44      {
45        Serial.println("NO sygnal for firstByte");
46      }
47
48      if(Wire.available())
49      {
50        secondByte = Wire.read();// get second byte!
51        Serial.print("secondByte");
52        Serial.println(secondByte);
53      }
54      else
55      {
56        Serial.println("NO sygnal for secondByte");
57      }
58  (...)
59  }
```

2 Aufgabe 2

Programmieren Sie das Launchpad bzw. den DS1621 in der oben gezeigten Schaltung dahingehend, dass der DS1621 kontinuierlich die aktuelle Umgebungstemperatur misst und das Launchpad diese über den Serial Monitor ausgibt.
Programmieren Sie die Schaltung so, dass beim Überschreiten einer einstellbaren Temperatur (z.B. 24 °C) die LED an und beim Unterschreiten der Temperatur die LED ausgeht.

Code:

```
1  // DS1621 Registers & Commands
2  /*
3  #define RD_TEMP     0xAA                    // read
       temperature register
4  #define ACCESS_TH   0xA1                    // access high
       temperature register
5  #define ACCESS_TL   0xA2                    // access low
       temperature register
6  #define ACCESS_CFG  0xAC                    // access
       configuration register
7  #define RD_CNTR     0xA8                    // read counter
       register
8  #define RD_SLOPE    0xA9                    // read slope
       register
9  #define START_CNV   0xEE                    // start
       temperature conversion
10 #define STOP_CNV    0X22                    // stop
       temperature conversion
11
12 // DS1621 configuration bits
13
14 #define DONE        B10000000               // conversion
       is done
15 #define THF         B01000000               // high temp
       flag
16 #define TLF         B00100000               // low temp flag
17 #define NVB         B00010000               // non-volatile
       memory is busy
18 #define POL         0x02                    // output
       polarity (1 = high, 0 = low)
19 #define ONE_SHOT    0x01                    // 1 = one
       conversion; 2 = continuous conversion
20 */
21 void setup()
22 {
```

```
23      Serial.begin(9600);
24      Wire.setModule(0);
25
26      Wire.begin();
27      Wire.beginTransmission(DEV_ID);                 // connect to
            DS1621 (#0)
28      Wire.write(0xAC);                               // Access Config
29      Wire.write(0x2);                                // set for
            continuous conversion
30      Wire.endTransmission();
31      Wire.beginTransmission(DEV_ID);                 // restart
32      Wire.write(0xEE);                               // start
            conversions
33      Wire.endTransmission();
34      // Tout ist aktiv wenn TH = 28.0° überschritten wird und
            setzt sich zurück, sobald der
35      // Wert unter TL= 26.0° fällt
36      Wire.beginTransmission(DEV_ID);                 // restart
37      Wire.write(0xA1); // change TH
38      Wire.write(0x1C); // Wert 28
39      Wire.write(0x0);  // Wert 0
40      Wire.endTransmission();
41
42      Wire.beginTransmission(DEV_ID);               // restart
43      Wire.write(0xA2); // change TL
44      Wire.write(0x1A); // Wert 26
45      Wire.write(0x0);  // Wert 0
46      Wire.endTransmission();
47  }
48
49  void loop()
50  {
51      int8_t firstByte = 0;
52      int8_t secondByte = 0;
53      float result = 0;
54
55      delay(3000);// give time (3 sec) for measurement
56
57      Wire.beginTransmission(DEV_ID);
58      Wire.write(0xAA);// read temperature command
59      Wire.endTransmission();
60
61      Wire.requestFrom(DEV_ID, 2);// ATTENTION: request from
            DS1621 (#0) two bytes (0.5 deg. resolution)
62
63      if(Wire.available())
64      {
65        firstByte = Wire.read();// get first byte!
66        Serial.print("firstByte");
```

```
67          Serial.println(firstByte);
68        }
69        else
70        {
71          Serial.println("NO sygnal for firstByte");
72        }
73
74        if(Wire.available())
75        {
76          secondByte = Wire.read();// get second byte!
77          Serial.print("secondByte");
78          Serial.println(secondByte);
79        }
80        else
81        {
82          Serial.println("NO sygnal for secondByte");
83        }
84
85        result = firstByte;
86
87        if (secondByte) // if there is a 0.5 deg difference
88            result += 0.5;
89
90        Serial.print("RESULT: ");
91        Serial.println(result);
92  }
```