

HN4-FTKL

Abteilung Elektronik

an der Höheren technischen Bundeslehranstalt 1
Innsbruck, Anichstraße 26 – 28

AquadDoc

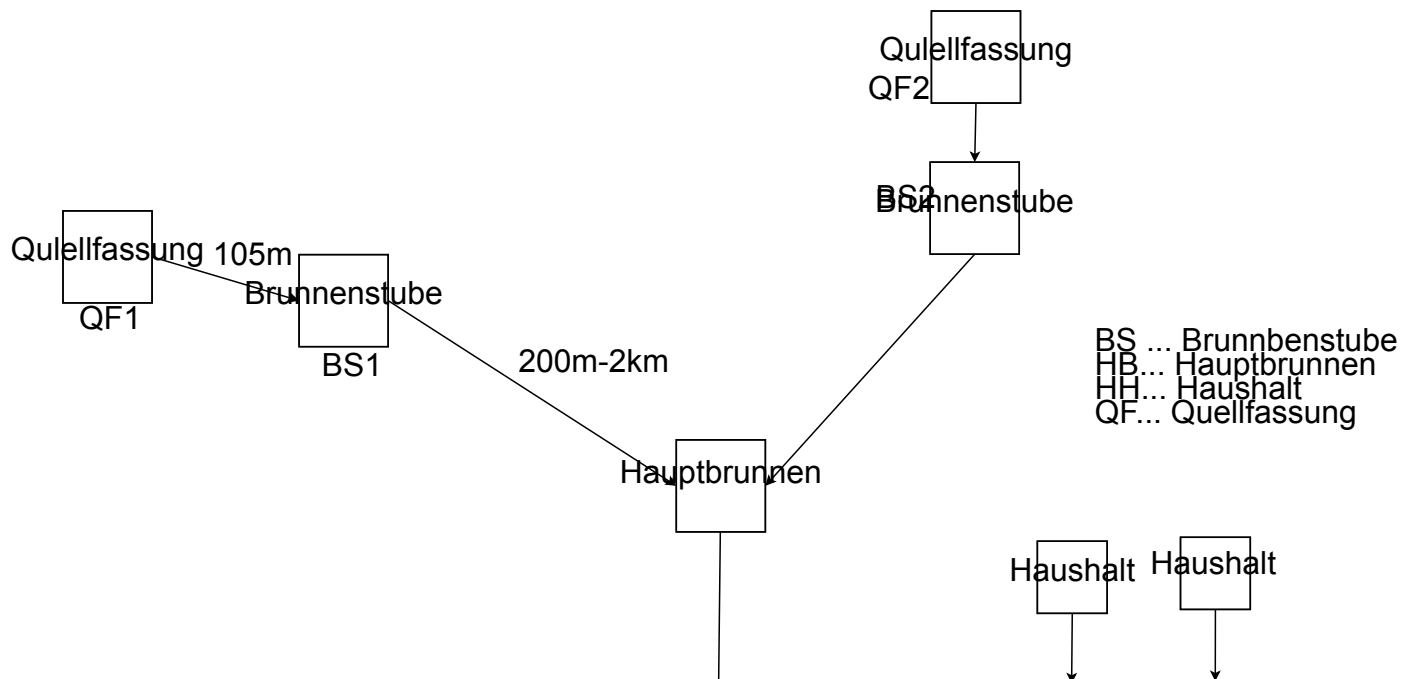
Manuel Ljubic, Jack Neuner, Daniel Plank

18. Februar 2019

Dokumentation des Wasserhaushalts einer Wasserversorgungsanlage für Kleinsiedlungsgebiete

Inhaltsverzeichnis

1	Aufgabenstellung	2
1.1	Aufbau	2
1.2	Sensoren	2
2	Test mittels Arduino	3
2.1	HC-SR04	4
2.2	DS1722	5
3	Addendum	7
3.1	Literaturverzeichnis	7



1 Aufgabenstellung

1.1 Aufbau

Ein Hochbehälter hat ein Fassungsvermögen um 1 Tag Wasser speichern zu können und ein Feuer zu löschen.

1.2 Sensoren

- Wasser zwischen Quelle und Brunnen und Brunnen und Häuser
Umsetzung mittels Rotor mit Magnet an einer Schaufel, welcher einen REED-Kontakt schaltet. Aufgabe: Prellt dieser Schalter?

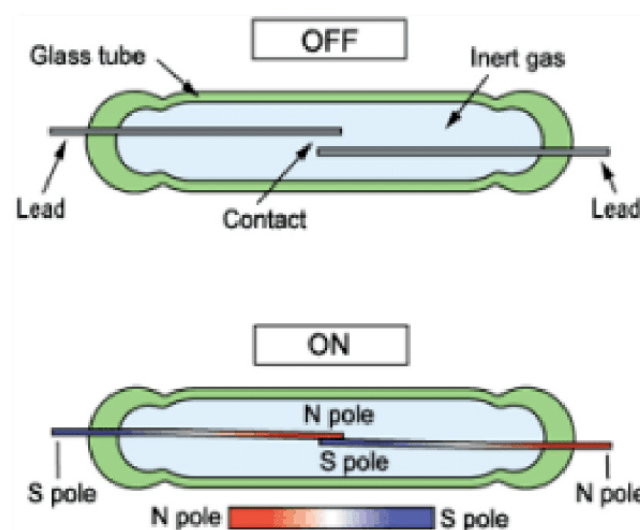


Abbildung 1: Schemata eines Reed-Schalters

- Wassertemperatur
Digitaler Sensor → Auflösung von 0.01°C

- Wasserstand Füllstand HB: Drucksensor, Ultraschallsensor, Potentiometer mit Schwimmer, etc.
- Türschalter
- Batteriestandsanzeige

Anforderungen an die Funkübertragung:

- Datensicherheit
- Übertragungsmöglichkeit (Manchester Kodierung)
- Übertragungsantenne

Gefundene Bauteile auf neuhold-elektronik.at:

- DS1722:
SPI Digital Thermometer. 8-12 bit Auflösung.
- HC - SR04:
PWM Ultraschall Messmodul. 2mA standby strom.
- MAX640:
5V Step-Down DC-DC Converter.

2 Test mittels Arduino

Es wird ein Arduino verwendet um die gegebenen Sensoren DS1722 bzw. HC-SR04 zu testen. Die SPI anschlüsse des Arduinos werden mit den Anschlüssen der jeweiligen Sensoren verbunden und auf die Verwendung der richtigen Versorgungsspannung wird geachtet.

2.1 HC-SR04

Listing 1: Arduino Programm für HC-SR04

```
1  /*
2  * Ultrasonic Sensor HC-SR04 and Arduino Tutorial
3  *
4  * by Dejan Nedelkovski,
5  * www.HowToMechatronics.com
6  *
7  */
8
9  // defines pins numbers
10 const int trigPin = 9;
11 const int echoPin = 10;
12
13 // defines variables
14 long duration;
15 int distance;
16
17 void setup() {
18     pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
19     pinMode(echoPin, INPUT); // Sets the echoPin as an Input
20     Serial.begin(9600); // Starts the serial communication
21     return;
22 }
23
24 void loop() {
25     // Clears the trigPin
26     digitalWrite(trigPin, LOW);
27     delayMicroseconds(2);
28
29     // Sets the trigPin on HIGH state for 10 micro seconds
30     digitalWrite(trigPin, HIGH);
31     delayMicroseconds(10);
32     digitalWrite(trigPin, LOW);
33
34     // Reads the echoPin, returns the sound wave travel time
35     // in  $\mu$ S
36     duration = pulseIn(echoPin, HIGH);
37
38     // Calculating the distance
39     distance= duration*0.034/2;
40
41     // Prints the distance on the Serial Monitor
42     Serial.print("Distance: ");
43     Serial.println(distance);
44     return;
45 }
```

2.2 DS1722

Listing 2: Arduino Programm für DS1722

```

1  #define DATAOUT 11 //MOSI – Master Input Slave Output
2  #define DATAIN 12 //MISO – Master Output Slave Input
3  #define SPICLOCK 13 //SCK — Serial Clock
4  #define SLAVESELECT 10 //SS — Slave Select
5  #define DS1722_POWER 9
6
7  #define DS1722_SELECT HIGH
8  #define DS1722_DESELECT LOW
9
10 #define DS1722_CONFIG_BYTE 0xEE
11 #define CONFIG_REG_READ 0x00
12 #define CONFIG_REG_WRITE 0x80
13 #define TEMP_ADDR_HI 0x02
14 #define TEMP_ADDR_LOW 0x01
15
16 byte clr;
17 byte temperature[2];
18
19 char spi_transfer(volatile char data){
20
21     // Start the transmission
22     SPDR = data; // Wait the end of the transmission
23     while (!(SPSR & (1<<SPIF))){};
24     // return the received byte
25     return SPDR;
26 }
27
28 void setup(){
29
30     byte n, config = 0xAB;
31     Serial.begin(9600);
32
33     temperature[0] = 0x12;
34     temperature[1] = 0x34;
35
36     /* Set DDIR registers */
37
38     pinMode(DATAOUT, OUTPUT);
39     pinMode(DATAIN, INPUT);
40     pinMode(SPICLOCK, OUTPUT);
41     pinMode(SLAVESELECT, OUTPUT);
42     pinMode(DS1722_POWER, OUTPUT);
43
44     digitalWrite(DS1722_POWER, HIGH); //disable device
45     delay(250);
46

```

```
47     digitalWrite(SLAVESELECT, DS1722_DESELECT); //disable
48         device
49     // set up SPI control register
50     SPCR = (1<<SPE)|(1<<MSTR)|(1<<CPOL)|(1<<CPHA);
51     clr=SPSR;
52     clr=SPDR;
53     delay(10);
54
55     // read config byte
56     digitalWrite(SLAVESELECT, DS1722_SELECT);
57     spi_transfer(CONFIG_REG_READ);
58     config = spi_transfer(0xFF);
59     digitalWrite(SLAVESELECT, DS1722_DESELECT);
60     delay(100);
61
62     Serial.print(config, HEX);
63     Serial.print('\n', BYTE);
64
65     // write config byte to the configuration register
66     digitalWrite(SLAVESELECT, DS1722_SELECT);
67     spi_transfer(CONFIG_REG_WRITE);
68     spi_transfer(DS1722_CONFIG_BYTE);
69     digitalWrite(SLAVESELECT, DS1722_DESELECT);
70     delay(100);
71
72     // read config byte
73     digitalWrite(SLAVESELECT, DS1722_SELECT);
74     spi_transfer(CONFIG_REG_READ);
75     config = spi_transfer(0xFF);
76     digitalWrite(SLAVESELECT, DS1722_DESELECT);
77     delay(100);
78
79     Serial.print(config, HEX);
80     Serial.print('\n', BYTE);
81
82     Serial.print('h', BYTE);
83     Serial.print('i', BYTE);
84     Serial.print('\n', BYTE); //debug
85     delay(1000);
86
87     return;
88 }
89 void loop(){
90
91     //float c, f;
92     // write config byte to the configuration register
93     digitalWrite(SLAVESELECT, DS1722_SELECT);
94     spi_transfer(CONFIG_REG_WRITE);
95     spi_transfer(DS1722_CONFIG_BYTE);
```

```
96     digitalWrite(SLAVESELECT, DS1722_DESELECT);
97
98     delay(1400);
99
100    digitalWrite(SLAVESELECT, DS1722_SELECT);
101    spi_transfer(TEMP_ADDR_HI);
102    temperature[0] = spi_transfer(0x00);
103    //release chip, signal end transfer
104    digitalWrite(SLAVESELECT, DS1722_DESELECT);
105    delay(25); // just because....
106    digitalWrite(SLAVESELECT, DS1722_SELECT);
107    spi_transfer(TEMP_ADDR_LOW);
108    temperature[1] = spi_transfer(0x00);
109    //release chip, signal end transfer
110    digitalWrite(SLAVESELECT, DS1722_DESELECT);
111    Serial.print(temperature[0] * 9 / 5 + 32, DEC);
112    Serial.print(' ', BYTE);
113    Serial.print(temperature[0], DEC);
114
115    if(temperature[1] & 0x80){
116
117        Serial.print(' ', BYTE);
118        Serial.print('5', BYTE);
119    }
120
121    Serial.print('\r', BYTE);
122    Serial.print('\n', BYTE);
123
124    delay(2000);
125
126    return;
127 }
```

3 Addendum

3.1 Literaturverzeichnis

- HC-SR04 Datenblatt
- DS1722 Datenblatt
- MAX640 Datenblatt
- CY8C5888LTI-LP097 Datenblatt