

F_Thingspeak AUSSEN OG Temperaturmessung V1_04.ino

```
1 /*
2  TEMPERATURMESSUNG INNEN/AUSSEN über Thingspeak
3  V1.00 23.06.2015 Neues Programm
4  V1.01 24.06.2015 Senden von 1 Dezimalstelle in Temperatur (z.B. 10.5), Abgleich mit TCM Thermometer (TCM Thermometer zeigt nun 0.1°C mehr an)
5  V1.02 26.06.2015 Erweiterung des Programms für 2 Varianten Innen/Aussen
6  V1.03 23.07.2015 Änderungen sodass Thingspeak Update wieder funktioniert - Nichtfunktion dürfte an Funktion StartEthernet gelegen sein
7  V1.04 01.08.2015 Umstellung auf Meldung an Thingspeak bei Min Intervall von 0,2
8
9  Arduino --> ThingSpeak Channel via Ethernet
10
11 The ThingSpeak Client sketch is designed for the Arduino and Ethernet.
12 This sketch updates a channel feed with an analog input reading via the
13 ThingSpeak API (https://thingspeak.com/docs)
14 using HTTP POST. The Arduino uses DHCP and DNS for a simpler network setup.
15 The sketch also includes a Watchdog / Reset function to make sure the
16 Arduino stays connected and/or regains connectivity after a network outage.
17 Use the Serial Monitor on the Arduino IDE to see verbose network feedback
18 and ThingSpeak connectivity status.
19
20 Getting Started with ThingSpeak:
21
22 * Sign Up for New User Account - https://thingspeak.com/users/new
23 * Create a new Channel by selecting Channels and then Create New Channel
24 * Enter the Write API Key in this sketch under "ThingSpeak Settings"
25
26 Arduino Requirements:
27
28 * Arduino with Ethernet Shield or Arduino Ethernet
29 * Arduino 1.0+ IDE
30
31 Network Requirements:
32 * Ethernet port on Router
33 * DHCP enabled on Router
34 * Unique MAC Address for Arduino
35
36 Created: October 17, 2011 by Hans Scharler (http://www.nothers.com)
37
38 Additional Credits:
39 Example sketches from Arduino team, Ethernet by Adrian McEwen
40
41 */
42
43 #include <SPI.h>
```

```

44 #include <Ethernet.h>
45 #include <OneWire.h>
46 #include <DallasTemperature.h>
47 #define ONE_WIRE_BUS 9 // Pin 9 ist der Sensor pin
48
49 // Setup a oneWire instance to communicate with any OneWire devices (not just Maxim/Dallas tem
perature ICs)
50 OneWire oneWire(ONE_WIRE_BUS);
51
52 // Pass our oneWire reference to Dallas Temperature.
53 DallasTemperature sensors(&oneWire);
54
55 // Local Network Settings
56 byte mac[] = { 0xD4, 0x28, 0xB4, 0xFF, 0xA0, 0xA3 }; // Must be unique on local network
57
58 // ThingSpeak Settings
59 char thingSpeakAddress[] = "api.thingspeak.com";
60 String writeAPIKey = "Hier eigenen API Key aus Thingspeak einfügen!";
61 #define SCHWELLE 0.2 // Min-Unterschied zwischen zwei Messwerten damit auf Thingspeak geschr
ieben wird
62 #define WARTEN 10000 // Wartezeit zw. zwei Messungen
63 #define updateThingSpeakInterval 300000 // Time interval in milliseconds to update ThingSpeak
(number of seconds * 1000 = interval) 300s = 5 min
64 #define maxThingSpeakInterval 1800000 // Max Time interval in milliseconds to update ThingSpea
k (number of seconds * 1000 = interval) 1800s = 30 min
65
66 // Variable Setup
67 long lastConnectionTime = 0;
68 boolean lastConnected = false, beginn = true;
69 int failedCounter = 0;
70 float Temp, Tempold = 0;
71 String TempStr;
72 char analogValue0[9];
73
74 // Initialize Arduino Ethernet Client
75 EthernetClient client;
76
77 void setup()
78 {
79 // Phicomm M1 Router Zeit zum starten geben
80 delay (50000);
81
82 // Start Serial for debugging on the Serial Monitor
83 // Serial.begin(9600);
84
85 lastConnectionTime = millis();
86
87 sensors.begin(); // IC Default 9 bit. If you have troubles consider upping it 12. Ups the de
lay giving the IC more time to process the temperature measurement
88 Serial.println (F("Sensors initialized."));
89

```

```

90 // Start Ethernet on Arduino
91 startEthernet();
92 }
93
94 void loop()
95 {
96 // call sensors.requestTemperatures() to issue a global temperature
97 // request to all devices on the bus
98 delay (WARTEN);
99 sensors.requestTemperatures(); // abfragen der Temperatur
100 Temp = sensors.getTempCByIndex(0); //lesen der Temperatur, why "byIndex"? You can have more
    than one IC on the same bus. 0 refers to the first IC on the wire
101 dtostrf(Temp, 7, 1, analogValue0);
102
103 // Print Update Response to Serial Monitor
104 while (client.available())
105 {
106     char c = client.read();
107     Serial.print(c);
108 }
109
110 // Disconnect from ThingSpeak
111 if (!client.connected() && lastConnected)
112 {
113     Serial.println("...disconnected");
114     Serial.println();
115
116     client.stop();
117 }
118
119 if (lastConnectionTime > millis()) { // Überlauf alle 50 Tage
120     lastConnectionTime = millis(); beginn = true;
121 }
122 // Update ThingSpeak
123 if(!client.connected() && (((millis() - lastConnectionTime > updateThingSpeakInterval))) ||
    (beginn == true)) {
124     if (((abs(Temp - Tempold) >= SCHWELLE) || (beginn == true) || (millis() - lastConnectionTi
    me > maxThingSpeakInterval)) && (Temp > -60)) {
125         TempStr = "field1=" + String(analogValue0) + " "; //field 1 ist das Thingspeak field
126         Serial.print ("*** Temp = "); Serial.println (TempStr);
127         updateThingSpeak(TempStr);
128         beginn = false;
129         Tempold = Temp;
130     }
131     else {
132         Serial.print ("Temp = "); Serial.println (TempStr);
133     }
134 }
135 else {
136     Serial.print ("Temp = "); Serial.println (TempStr);
137 }

```

```

138
139 // Check if Arduino Ethernet needs to be restarted
140 if (failedCounter > 3 ) {startEthernet();}
141
142 lastConnected = client.connected();
143 }
144
145 void updateThingSpeak(String tsData)
146 {
147   if (client.connect(thingSpeakAddress, 80))
148   {
149     client.print("POST /update HTTP/1.1\n");
150     client.print("Host: api.thingspeak.com\n");
151     client.print("Connection: close\n");
152     client.print("X-THINGSPEAKAPIKEY: "+writeAPIKey+"\n");
153     client.print("Content-Type: application/x-www-form-urlencoded\n");
154     client.print("Content-Length: ");
155     client.print(tsData.length());
156     client.print("\n\n");
157
158     client.print(tsData);
159
160     lastConnectionTime = millis();
161
162     if (client.connected())
163     {
164       Serial.println("Connecting to ThingSpeak...");
165       Serial.println();
166
167       failedCounter = 0;
168     }
169     else
170     {
171       failedCounter++;
172
173       Serial.println("Connection to ThingSpeak failed ("+String(failedCounter, DEC)+")");
174       Serial.println();
175     }
176
177   }
178   else
179   {
180     failedCounter++;
181
182     Serial.println("Connection to ThingSpeak Failed ("+String(failedCounter, DEC)+")");
183     Serial.println();
184
185     lastConnectionTime = millis();
186   }
187 }
188

```

```
188
189 void startEthernet()
190 {
191
192   client.stop();
193
194   Serial.println(F("Connecting Arduino to network..."));
195
196   delay(1000);
197
198   // Connect to network and obtain an IP address using DHCP
199   if (Ethernet.begin(mac) == 0)
200   {
201     Serial.println(F("DHCP Failed, reset Arduino to try again"));
202     Serial.println();
203   }
204   else
205   {
206     Serial.println(F("Arduino connected to network using DHCP"));
207     Serial.println();
208   }
209
210   delay(1000);
211 }
```