

hier unverbindlich und ohne Gewähr eine Zusammenfassung der Erfahrung mit den hygrosens-Modulen zur Ermittlung der relativen Feuchtigkeit / Temperatur

TWI-Adresse

Wenn man sich die TWI-Interna anschaut, dann besteht eine TWI-/I2C-Adresse aus 7 Bit mit angehängtem Read/Write Bit.

Insgesamt ist sie also eigentlich 8 Bit = 1 Byte lang.

Das R/W-Bit entscheidet, ob - aus der Sicht des Masters - eine Schreibvorgang (R/W = 0) oder ein Lesevorgang (R/W = 1) durchgeführt werden soll.

Bit.7	Bit.6	Bit.5	Bit.4	Bit.3	Bit.2	Bit.1	Bit.0
A.6	A.5	A.4	A.3	A.2	A.1	A.0	R/W

Die HYT-xxx haben die 7-Bit Adresse 0b0101000 (könnte man lesen als 0x28).

Diese Adresse muss aber nach links geschoben werden, damit an der richtigen Stelle im niedrigstwertigen Bit Platz für das Read/Write Bit bleibt.

Bit.7	Bit.6	Bit.5	Bit.4	Bit.3	Bit.2	Bit.1	Bit.0
0	1	0	1	0	0	0	R/W

Als Hexadezimalwert könnte man hier lesen 0x50 (wenn das R/W Bit nicht gesetzt ist) bzw. 0x51 (wenn das R/W-Bit gesetzt ist).

Verwirrung ist entstanden, weil einige TWI-Modul die TWI-Adresse in der 7-Bit Darstellung erwartet. Hier wird intern die Adresse um 1 Bit nach links geschiftet um ein R/W-Bit anhängen zu können.

Andere erwarten die Adresse direkt in der 8-Bit-Darstellung.

Daten aus dem Modul auslesen

Um Daten auszulesen, schickt man zuerst die TWI-Adresse mit R/W-Bit = 0 an das Modul.

Bit.7	Bit.6	Bit.5	Bit.4	Bit.3	Bit.2	Bit.1	Bit.0
0	1	0	1	0	0	0	0

Zur Kontrolle sollte man prüfen, ob der Slave (das Modul) mit einem ACK antwortet (er zieht kurz die SDA-Leitung auf Masse und lässt sie dann wieder "los").

Ist keine Reaktion an SDA zu erkennen, dann liegt ein Fehler vor, der beseitigt werden muss.

Wird ein ACK empfangen, dann muss man dem Modul Zeit gewähren, um die Sensoren für Feuchtigkeit und Temperatur auszulesen, in kalibrierte Werte umzurechnen und das Ergebnis in seinem Ausgangspuffer bereitzustellen (ca. 100ms).

Anschließend wird die TWI-Adresse mit gesetztem R/W-Bit gesendet:

Bit.7	Bit.6	Bit.5	Bit.4	Bit.3	Bit.2	Bit.1	Bit.0
0	1	0	1	0	0	0	1

und direkt vier Byte aus dem Modul ausgelesen.
Diese Bytes mögen in einem Array namens Buffer[] abgelegt sein.

Daten interpretieren

In Buffer[0] signalisieren die beiden höchstwertigen Bits:

Bit.7	Bit.6	Bit.5	Bit.4	Bit.3	Bit.2	Bit.1	Bit.0
CmdMode	Busy	x	x	x	x	x	x

Bit.7 sollte nie gesetzt sein, sofern man nicht absichtlich den 'Command Mode' aktiviert hat.
Ein gesetztes Bit.6 zeigt an, dass das Modul noch keine neuen Daten bereitgestellt hat (z.B., weil es noch mit der Konversion beschäftigt ist).

Sind weder Bit.7 noch Bit.6 gesetzt, dann können man die Daten ausgewertet werden:

Raw-Daten für Relative Feuchtigkeit

Die beiden Bytes müssen zu einem 16Bit-Wert zusammengefügt werden.
Dazu benötigt man eine entsprechende Variable vom Typ unsigned int.

$raw_RH16 = Buffer[0] * 256 + Buffer[1]$; oder
 $raw_RH16 = Buffer[0] < 8 | Buffer[1]$;

Beide Schreibweisen führen zum gleichen Ergebnis.

Raw-Daten für Temperatur

Es gilt dieselbe Überlegung wie bei der relativen Luftfeuchtigkeit, die Bytes Buffer[2] und Buffer[3] werden zu einem 16Bit-Wert in der Variablen T16 zusammengeführt:

$raw_T16 = Buffer[2] * 256 + Buffer[3]$; oder
 $raw_T16 = Buffer[2] < 8 | Buffer[3]$;

Der Hersteller empfiehlt, die beiden niedrigwertigen Bits zu entfernen (maskieren).
In einigen Beispielen wird die Variable anschließend noch 2x nach rechts geschiftet (also durch 4 dividiert). Hier liegt aber ein Missverständnis vor, das ist nicht erforderlich !

Umrechnung der Raw-Daten

Die Feuchtigkeit wird als Wert zwischen 0 ... 2¹⁴ übermittelt und muss umgerechnet werden auf den Wertebereich 0 ... 100%:

$RH = raw_RH16 * 100 / 2^{14}$

Bei der Umrechnung der Temperatur scheinen sich die Module zu unterscheiden.

Für den HYT-221 gilt:

Die Temperatur wird als Wert zwischen 0 ... 2^{14} übermittelt und muss umgerechnet werden nach der Vorschrift:

$$T = ((\text{raw_T16} * 165) / 2^{14}) - 40$$

Für den HYT-271 und möglicherweise auch andere Module gilt:

Die Temperatur wird als Wert zwischen 0 ... 2^{16} übermittelt und muss umgerechnet werden nach der Vorschrift:

$$T = ((\text{raw_T16} * 165) / 2^{16}) - 40$$

Wer mit einem 8-Bit Microcontroller arbeitet, der sollte die Umrechnungen nicht mit Fließkomma-Arithmetik sondern mit Ganzzahlen ausführen: das 'kostet' wesentlich weniger Speicherplatz.

06.05.11

Michael S.