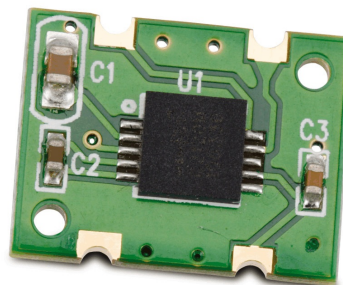


---

## Kompass-Modul HDMM01

---

Unter der Bezeichnung HDMM01 bietet die Firma Pollin einen Magnetfeldsensor für ca. 7 Euro an. Dieser Baustein lässt sich nicht nur zur Bereitstellung von Kompassfunktionen nutzen; mit ihm kann man auch magnetische Feldstärken messen.



Herzstück der kleinen Platine ist der Baustein **MMC2120M** der Firma MEMSIC. Es handelt sich dabei um einen Zwei-Achsen-Magnetfeldsensor, der mit dem GMR-Effekt (Riesen-Magneto-Widerstand) arbeitet. Leider findet man auf der Homepage der Firma kein Datenblatt (mehr) zu diesem Baustein. Die Firma Pollin liefert eine verkürzte Version; sie ist in manchen Aspekten aber lückenhaft. Ein ausführlicheres Datenblatt finden Sie aber in meinem Attiny-Forum unter <http://www.forum.g-heinrichs.de/viewtopic.php?f=12&t=86&p=117&hilit=HDMM01#p117>.

### Hier nun einige wichtige Details im Überblick:

Der Baustein kann mit **3 - 5 V** versorgt werden; er kann also direkt an die Attiny-Platine angeschlossen werden. Allerdings muss man dazu noch 4 Kabel anlöten (GND, VCC, SDA und SCL)

Die Kommunikation läuft über das **I2C-Protokoll**; die Adresse ist &H60 (Schreiben) bzw. &H61(Lesen).

Der Sensor kann magnetische Feldstärken im **Bereich von - 400 uT bis + 400 uT** messen. Überschreitet man diese Grenzen, kann es zu einer Ummagnetisierung des Sensormaterials kommen; der Sensor liefert dann falsche Ergebnisse. Durch eine Reset-Prozedur (über eine im Baustein eingebaute winzige Spule) kann dieses Problem aber behoben werden. Bei Feldstärken über 1 T kann der Sensor dauerhaft(!) zerstört werden.

Der Sensor liefert für die x- und die y-Komponente des Magnetfeldes Rohwerte (counts) im Bereich zwischen 0 und 4095 ( $=2^{12}-1$ ). Dabei gilt folgende Zuordnung:

0 counts -> -400 uT  
2048 counts -> 0 uT  
4096 count -> 400 uT

Um von den Rohwerten zu den Feldstärken (in uT) zu gelangen, muss man also von dem Countwert 2048 subtrahieren und das Ergebnis durch 5,12 dividieren.

Bei Mikrocontrollern wie den Attiny 2313, die wenig Programmspeicher besitzen, wird man häufig auf die Benutzung von Dezimalzahlen verzichten müssen. In diesem Fall kann man folgende Näherung für die Umskalierung benutzen:

---

## Kompass-Modul HDMM01

---

$b = \text{rohwert}/5$   
 $\text{korrektur} = b/50$   
 $b = b - \text{korrektur}$

Die Umskalierung ist damit schon wesentlich genauer als die vom Hersteller garantierte Genauigkeit der Messwerte.

Wie geht man nun mit dem Baustein um?

Der Baustein hat insgesamt 5 Register mit den Adressen 0 bis 4:

- 0: Internes Register
- 1: MSB X-Achse
- 2: LSB X-Achse
- 3: MSB Y-Achse
- 4: LSB Y-Achse

Das interne Register dient zur Kontrolle und Steuerung des Bausteins (s. u.). Von den MSB-Registern werden jeweils nur die niederwertigsten 4 Bit benutzt, da die Messwerte ja nur 12 Bit besitzen.

**Eine Messung wird nun folgendermaßen durchgeführt:**

1. I2C-Start
2. schreibe: Schreibadresse
3. schreibe: 0 (Adresse des internen Registers)
4. schreibe: 1 (schreibt Aufwachbefehl in das interne Register; veranlasst Messvorgang)
5. I2C-Stop
6. min. 5 ms warten (Messvorgang braucht Zeit!)
7. I2C-Start
8. schreibe: Schreibadresse
9. schreibe: 1 (Adresse des ersten Messwertes)
10. I2C-Stop
11. I2C-Start
12. schreibe: Leseadresse
13. lese, NACK (Inhalt des Registers 1 lesen - vgl. Schritt 9 - und nächste Adresse ansteuern)
14. lese, NACK (Inhalt des Registers 2 lesen und nächste Adresse ansteuern)
15. lese, NACK (Inhalt des Registers 3 lesen und nächste Adresse ansteuern)
16. lese, ACK (Inhalt des Registers 4 lesen)
17. I2C-Stop

Bevor Messungen vorgenommen werden - oder wenn der Baustein wegen einer Überlastung nicht mehr korrekt arbeitet - kann man den Baustein in den Ausgangszustand versetzen. Dazu

---

## Kompass-Modul HDMM01

---

muss man Strom durch die oben bereits angesprochene Spule fließen lassen, und zwar zuerst in die eine Richtung und dann in die andere:

### SET/RESET-Prozedur

0. nach Einschalten min. 10 ms warten
1. I2CStart
2. schreibe: Schreibadresse
3. schreibe: 0 (Adresse des internen Registers)
4. schreibe: 2 (SET-Befehl, Strom einschalten)
5. 1 ms warten
6. schreibe: 4 (RESET-Befehl, umgekehrte Stromrichtung einschalten)
7. 5 ms warten
8. I2C-Stop

Die Logik des Baustein sieht also so aus:

Nach der erfolgreichen Adressierung des Bausteins durch "schreibe: &H60" sendet man als nächstes eine Zahl, die vom Baustein in den Adresspointer geladen wird. Der nächste Schreib- oder Lesevorgang bezieht sich auf das Register mit dieser Adresse. Nur beim Register 0 hat man Schreib- und Lesezugriff, die restlichen kann man nur lesen. Befehle werden gegeben, indem man einzelne Bits des internen Registers auf 1 setzt. Man kann die einzelnen Bits auch zur Kontrolle auslesen: Sie werden vom Baustein nach Beendigung jeder befohlenen Aktion nämlich automatisch auf 0 gesetzt. Somit lässt sich in Erfahrung bringen, ob der Baustein einen Befehl bereits ausgeführt hat oder nicht.