# Inbetriebnahme einer Magellan 8200

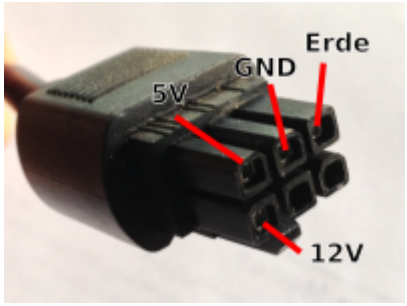
In jedem Discounter und Supermarkt gibt es mindestens sie: In das Kassensystem eingebaute stationäre Barcodescanner, die oft auch als Waage benutzt werden können. Das brauchen wir natürlich auch! Wir haben eine Magellan 8200 geschenkt bekommen und diese in Betrieb genommen.

Benötigte Teile:

- Magellan 8200 mit Standard-RS232 Schnittstelle
- Netzteil mit 5V/1,5A und 12V/1,5A
- 6poliger Molex-Stecker
- Ethernetkabel (CATx) mit RJ-45-Stecker
- Arduino Micro, Micro-USB-Kabel
- Schwarz-Weiß Drucker
- Handbuch

## Stromversorgung

Zur Stromversorgung werden 5V und 12V benötigt - idealerweise mit einem Netzteil welches beide Spannungen liefern kann. Da das gerade herumliegende Netzteil vermutlich nicht den passenden Stecker hat, muss noch ein 6poliger Molex-Stecker besorgt und volgendermaßen Verdrahtet werden:




Wenn das Netzteil jetzt eingeschaltet wird, sollte das System anlaufen. Je nach Konfiguration meldet es sich sogar mit einem Piepston. Die Sieben-Segment-Anzeige (7seg abgekürzt) sollte nur kurz aufleuchten. Wenn sie dauerhaft etwas anzeigt, dann ins handbuch auf Seite 80 den Fehler suchen.

## Herausfinden des Gerätetyps



Beim Barcode-Scannen von Seiten mit mehreren Barcodes darauf achten dass alle nicht benutzten Barcodes abgedeckt sind. Der Scanner scannt aus allen Richtungen!

Als nächstes muss herausgefunden werden, mit welchem Interface die Einheit angebunden werden kann. Dafür wird der Lautstärke-Knopf ca 5 Sekunden gedrückt, ein dreifacher schneller Piep sollte zu hören sein. Wenn nichts passiert außer dass bei 10sekündigem Tastendruck der Scanner neu startet, muss der Diagnosemodus erst aktiviert werden. Dafür müssen die Seiten 112 (SWITCH_LABEL) und 127 (SCANNER BUTTON OPTION = ENABLE ALL FUNCTIONS) ausgedruckt werden. Dann werden die

Barcodes in folgender Reihenfolge gescannt:

1. SWITCH
2. SCANNER BUTTON OPTION = ENABLE ALL FUNCTIONS
3. SWITCH

Die 7seg zeigt dann die Versionsnummer des Geräts an, in unserem Fall 1F-05 r96-978 Die 05 gibt bei uns das Interface RS-232 Standard an, bei anderen Zahlen hilft das Handbuch auf Seite 156 weiter - der hier dargestellte Weg funktioniert dann vermutlich nicht.

## Programmieren der seriellen Schnittstelle

Zur Benutzung muss die serielle Schnittstelle richtig konfiguriert sein. Zur Einfachheit wird hier nur der Barcode-Name und die Seite des Handbuchs benutzt, wenn etwas eingescannt werden soll. Als erstes wird der Werkszustand wiederhergestellt, um einen definierten Startzustand zu haben.

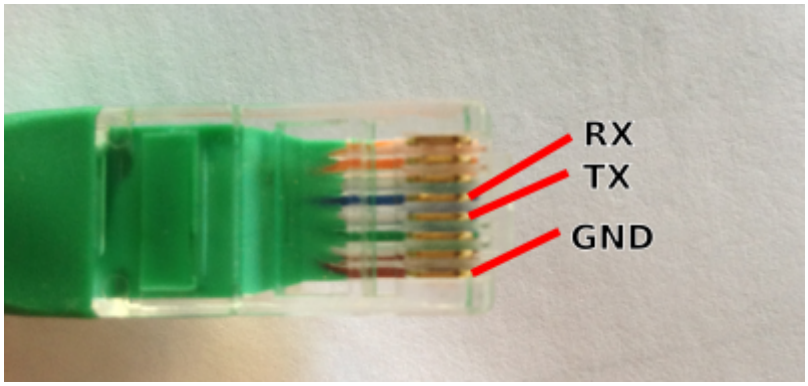
- RETURN TO FACTORY SETTINGS (117)

Damit ist der Diagnosemodus wieder deaktiviert und muss bei Bedarf wie oben aktiviert werden - wir brauchen ihn hier aber vorerst nicht mehr. Für alle weiteren Schritte muss immer zuerst der SWITCH LABEL(112) Barcode gescannt werden, dann der genannte Barcode und dann wieder SWITCH_LABEL - für jeden Schritt sind also drei Scanvorgänge nötig.

1. INTERFACE-Type = RS-232 STANDARD (158) Einstellen des Interfacetyps.
2. (GLOBAL_SUFFIX = CR LF (168) Sende jeden Barcode in einer neuen Zeile)
3. RS-232 BAUD RATE = 9600 (177) Setze die Baudrate auf 9600
4. RS-232 NUMBER OF DATA BITS = 8 (180)
5. RS-232 NUMBER OF STOP BITS = 1 (181)
6. RS-232 TTL INVERT = ENABLE (192)
7. RS-232 PARITY = EVEN (183) Check
8. RS-232 HARDWARE CONTROL = DISABLE (184)
9. RS-232 IGNORE HOST COMMANDS = DISABLE (190) Nötig?
10. RS-232 LEVELS = TTL (191) Überträgt die Signale mit 5V TTL-Level - WICHTIG!

## Testen mithilfe eines Arduino-Gerätes

Um einfach zu Testen ob alles funktioniert, wird ein Arduino Micro verwendet - andere 5V-kompatible Seriell-zu-USB-Wandler können natürlich ebenfalls verwendet werden, Verkabelung und Sketch müssen gegebenenfalls verändert werden. Zum Anschließen wird ein Ethernet-Kabel mit einem RJ-45 Steckverbinder an einem Ende benötigt. Folgende Adern werden benötigt:




- GND wird mit GND des Arduinos verbunden.
- RX wird mit TX des Arduinos verbunden.
- TX wird mit RX des Arduinos verbunden.

Auf den Arduino wird der Beispiel-Sketch MultiSerial geladen. Der RJ45-Stecker wird dann in den POS-Terminal-Eingang des Scanners eingesteckt und der Arduino per USB mit einem Computer verbunden. Dann wird der Scanner eingeschaltet und der Arduino-Serial-Monitor gestartet. Ein eingescannter Barcode sollte jetzt angezeigt werden. Wenn das Präfix des Barcodes stört, kann dies gegebenenfalls noch umprogrammiert werden. Folgende nützliche Kommandos können an den Scanner übertragen werden - einfach ins Sendefenster des Monitors eingeben:

- **B**: Piepton
- **F**: Dreifacher Piepton
- **E**: Schalte Scanner ein
- **D**: Schalte Scanner aus (Motor bleibt an, Laser geht aus)
- **i**: Sendet die ID des Scanners
- **h**: Sendet den Zustand des Scanners, antwort sollte „mOK hOK “ lauten
- **s**: Sendet den Status des Scanners, Lxx gibt an dass schon xx Barcodes gescannt wurden.

## Emulieren einer Tastatureingabe

Der Arduino Micro kann als tastatur verwendet werden. Die meisten Hand-Barcodescanner geben auch so ihre Daten weitergeben. Mit folgendem Arduino-Sketch ist das einfach möglich:

KeyboardScanner.ino

```
//Modified Example KeyboardSerial
#include "Keyboard.h"

void setup() {
  Serial1.begin(9600);
  Keyboard.begin();
}

void loop() {
  if (Serial1.available() > 0) {
    Keyboard.write(Serial1.read());
  }
```

```
}
```