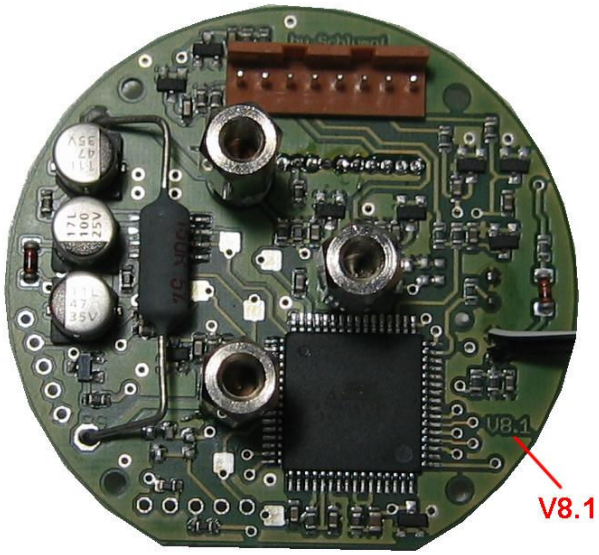
Overview

The Multi Gauge is an on-board computer for Yamaha FZS1000 (= FZ1 Gen1) motorbikes and the like. It also provides the option to connect external “add-ons” to enhance functionality.

Most popular is the use of a GPS receiver, the software is already capable to decode and display the data.

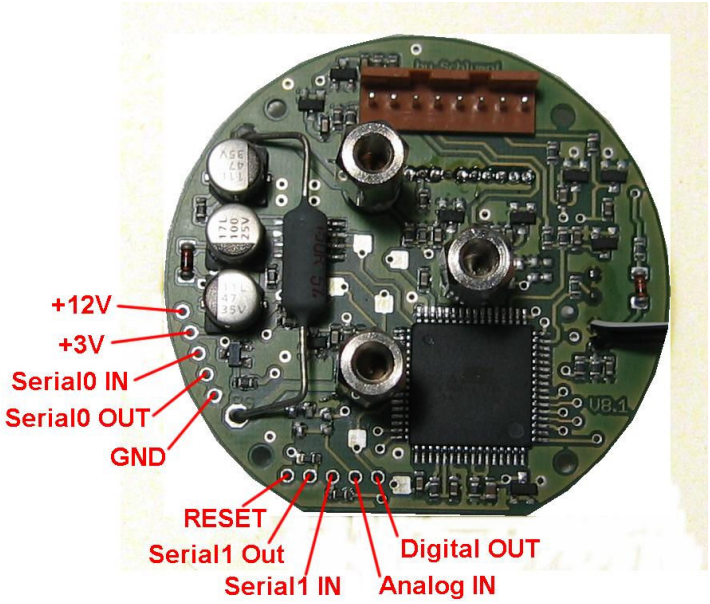
Hardware version

Before starting any modifications check the hardware version in place. The following explanantions only refer to version **V8.1**. The version number is etched into the copper. Older versions are electrically similar but do not offer dedicated solder points for easy access:



Extension headers

The following extension headers are intended to be used for external connections.



Adding a GPS receiver

The Multi Gauge is able to process a standard GPS-NMEA signal at its *serial0* input with the following properties:

- Data voltage level: 3V TTL, normal logic
- Baudrate: 1200 .. 38400 Baud
- Required messages: RMC, GGA

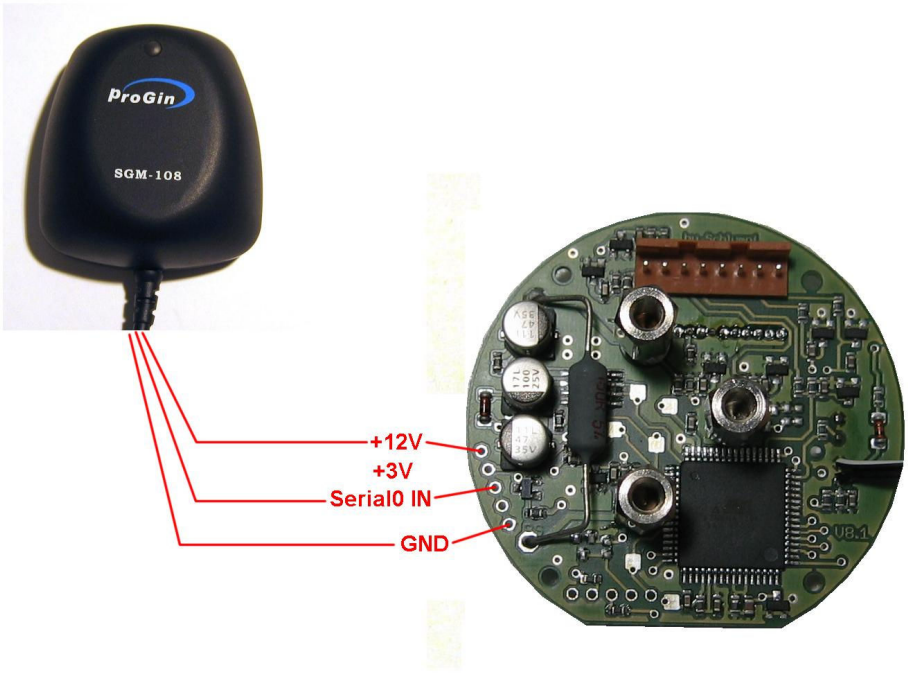
Those signals are standard for most GPS-modules.
RS-232 type modules are not directly usable due to invers logic and too high amplitudes.
Devices running at 3V power supply can not be used directly as the Multi Gauge's 3V is not capable to deliver sufficient currents.

Almost all GPS devices contain a (core) GPS module according to the above characteristics. In addition they have additional power supply, battery, USB or Bluetooth electronics, depending on application. With the right skills and tools it is possible to tap the respective signals. Using a bare GPS module instead is much more easy and save.

A cheap, though not easy, way to get GPS signals is to modify an old, cabled USB GPS receiver. Those devices typically run at 5V (due to USB). The cockpit provides a strong and stable 5V at CL8, see below, that serves very well as a voltage supply. Alternatively any 12V to 5V voltage regulator IC can be used.

- Located the USB 5V line in the receiver and connect it to CL8. Same for the ground line, connect it to a proper ground spot on the Multi Gauge or cockpit board.
- Now the difficult part. Find and tap the 3V NMEA TTL signal at the input of the USB-bridge IC (often PL2303). Route it to *Serial0 IN*.

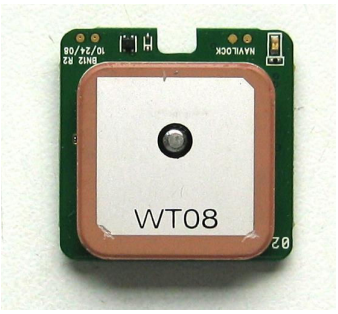
Connection to the multi gauge is done by 3 wires. As an example, a modified USB GPS receiver with additional 12V->5V regulator (my first attempt, it's more easy to use CL8):

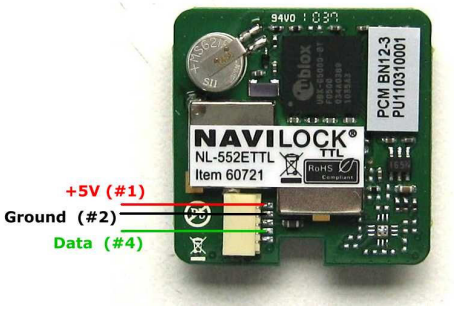


The NMEA signal goes to serial input #0 of the processor only. *Serial0 OUT* is not used as the multi gauge is a listen-only device.

GPS receivers also exist as bare modules, quite popular in the R/C community. Typical examples are "*Navilock NL-552ETTL*" or *OriginGPS ORG1318*. Those modules need no modification as the output is already of 3V TTL type.

Navilock NL-552ETTL:



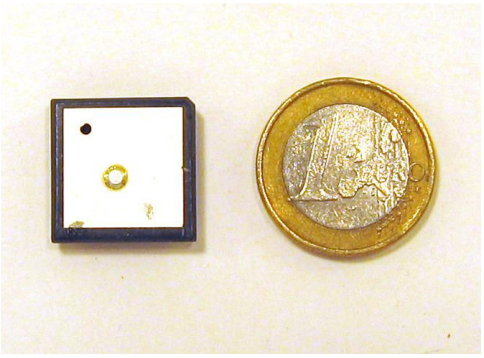


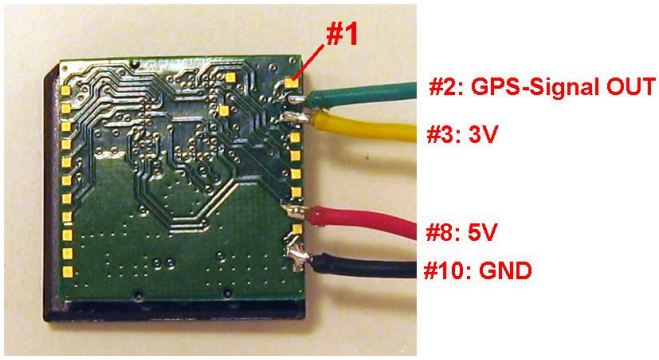
This module is my favorite as it's most simple to connect:

- 5V to CL8 on cockpit.
- Ground to *GND* on Multi Gauge.
- Data to *Serial0 IN* on Multi Gauge.

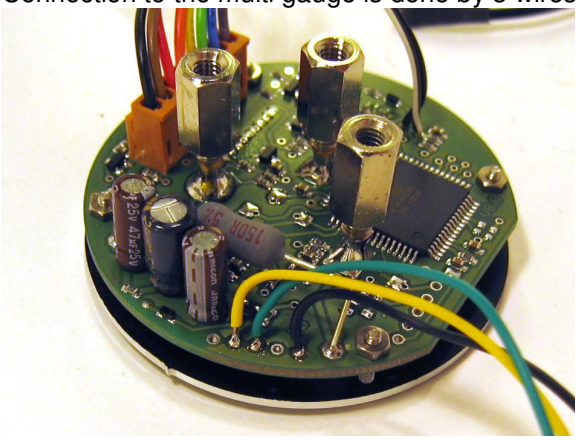
Take care to order exactly this type of module. It also exists in different flavors with USB (extension EUSB) or RS232 output (ERS), those are not usable without modifications!

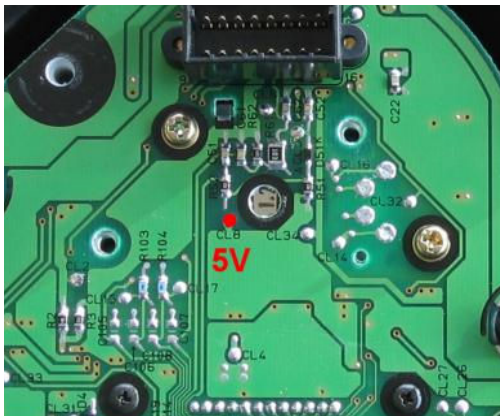
ORG1318:





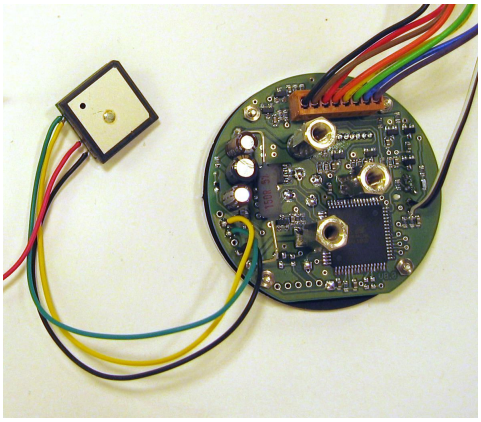
This module is very tiny. It needs an additional low current 3V supply at #3 as reference for the I/O signal level (the Multi Gauge is capable to deliver this).
Connection to the multi gauge is done by 3 wires, the remaining wire has to be connected to 5V (CL8):

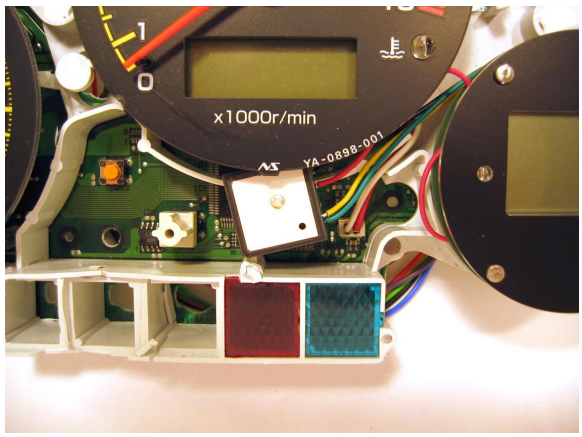




The most simple way to get a 5V supply is to tap the existing cockpit supply at "CL8" (or somewhere on the copper plane nearby). This supply is strong enough to also support an additional GPS module.

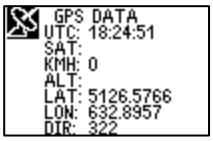
Using one of these tiny bare modules gives the great advantage to place it inside the cluster. This minimizes cabling and provides perfect protection against ambient conditions. The housing's plastic is transparent for GPS signals and gives ideal receiving conditions. For electrical isolation and precaution against vibrations the module should be wrapped into some air-bubble plastic foil or alike.

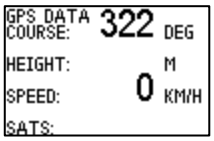


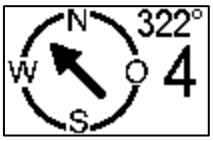


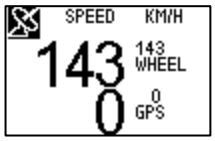
GPS support has to be enabled and set up in software in menu 2. Consult the user guide document for further details if needed. Enable, select baudrate and save. 4800 baud is a widely used standard. If set correctly message names will be shown in the upper right. Typically they are updated and altered every second:

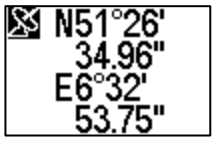
Viewmodes 30 up to 37 will start to display the GPS content. Depending on signal quality the antenna symbol (upper left corner) will be normal or inversed:

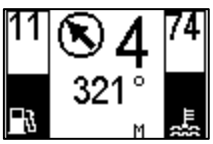


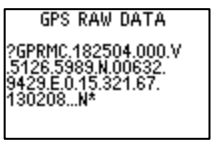












Connecting an external shift light LED

Terminal X2-4 (before named Analog IN, processor port PF5) can be used as output to control an external shift light. The output is active high and can deliver a few mA@3V. See schematic for details if an additional high current switching transistor is needed.

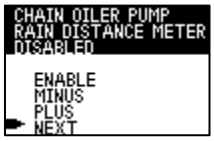
This function is activated by default.

Connecting a chain oiler pump

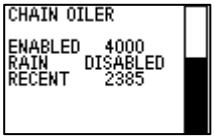
Terminal X2-5 (before named Digital OUT, processor port PB1) can be used as output to control a chain oiler pump. Originally this function was designed to work with a “McCoi” chain oiler system. It provides a short pulse at every 4000 meters (adjustable). The very first pulse distance is halved. RAIN mode can be set manually, distance is then set to 2000 meters (adjustable).

The multi gauge is in no way capable to carry the pump’s currents, a high current switching unit has to follow (e.g. MOSFET)!

This function has to be activated in menue 1. First menu sets normal distance (dry), second menu for rain:

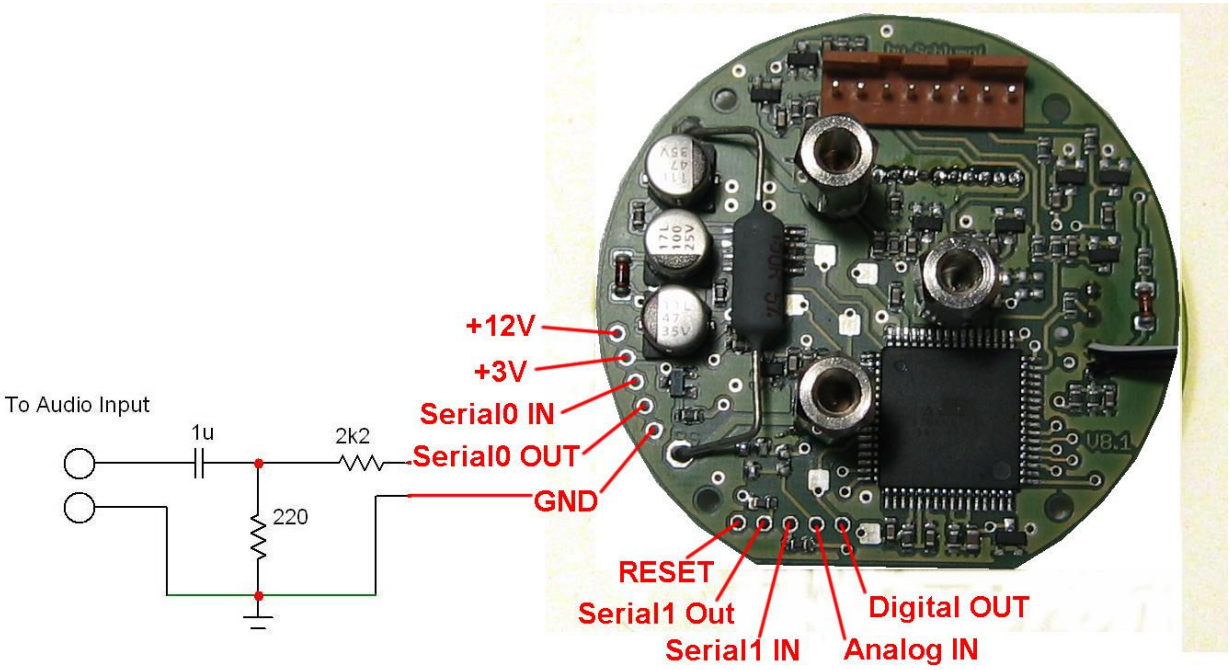


Viewmode 22 provides information about the actual status and elapsed distance:



Data Audio Streaming

The multi gauge is capable to provide all measurement values as an audio stream. The format is a kind of FSK, similar to early computer modem transmission. Storage can be done on any audio recorder, e.g. camcorder audio track or MP3 player. The data has to be post-processed on a PC to get it back in tabular format for subtitling or the like. A simple filter has to be added to the multi gauge and R22 shorted:



This function uses the same serial port as the GPS modification. It is not possible to run both at the same time.

Data audio streaming has to be enabled in menu 2. PLUS / MINUS shifts the fundamental frequencies:

The stream contains the following values, updated several times per second:

- Speed
- RPM
- Gear
- Elapsed distance
- Temperature water
- Temperature ambient
- Fuel level

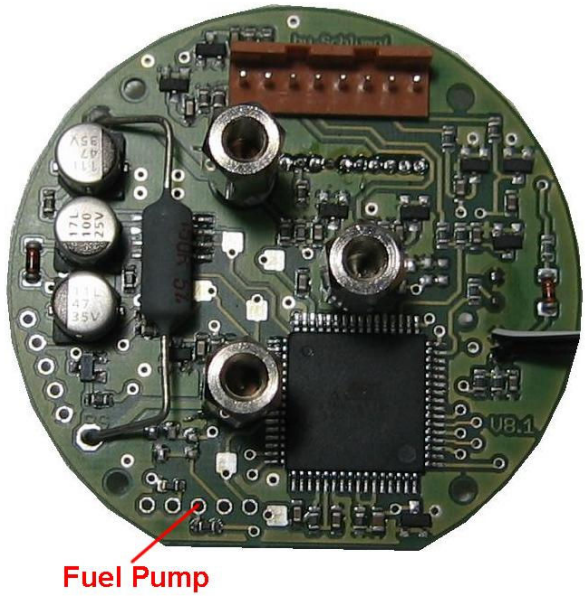
Also realtime subtiteling with e.g. MAX7456 is thinkable. This circuit is directly looped into the video signal. It gets the data from the multi gauge and adds the data as a subtitle into the video signal. Further informations on request.

Fuel Pump Connection

The Multi Gauge is capable to count the fuel pump switching pulses. By also taking the known, elapsed distance into account the gauge calculates the fuel consumption. This function is only supported by software versions from 807 onwards.

Due to the simple nature of this signal the achievable accuracy is limited.

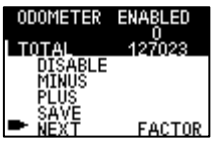
The fuel pump has to be connected to the point given in the following picture (X2-3). No further components are necessary, the signal conditioning and protecting elements are already placed on the board:



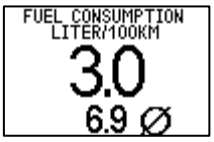
A new wire has to be routed (the blue one from the picture). On the fuel pump side the wire has to be connected to the black wire coming out of the pump housing.

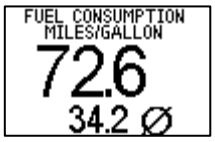
The fuel pump signal recognition has to be enabled and configured in menu 2. The number of pump-ticks per liter has to be set and fine tuned individually:

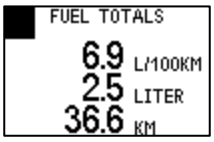
To preserve all data during ignition off the odometer-save-function has to be enabled:



Results are displayed with the following viewmodes:

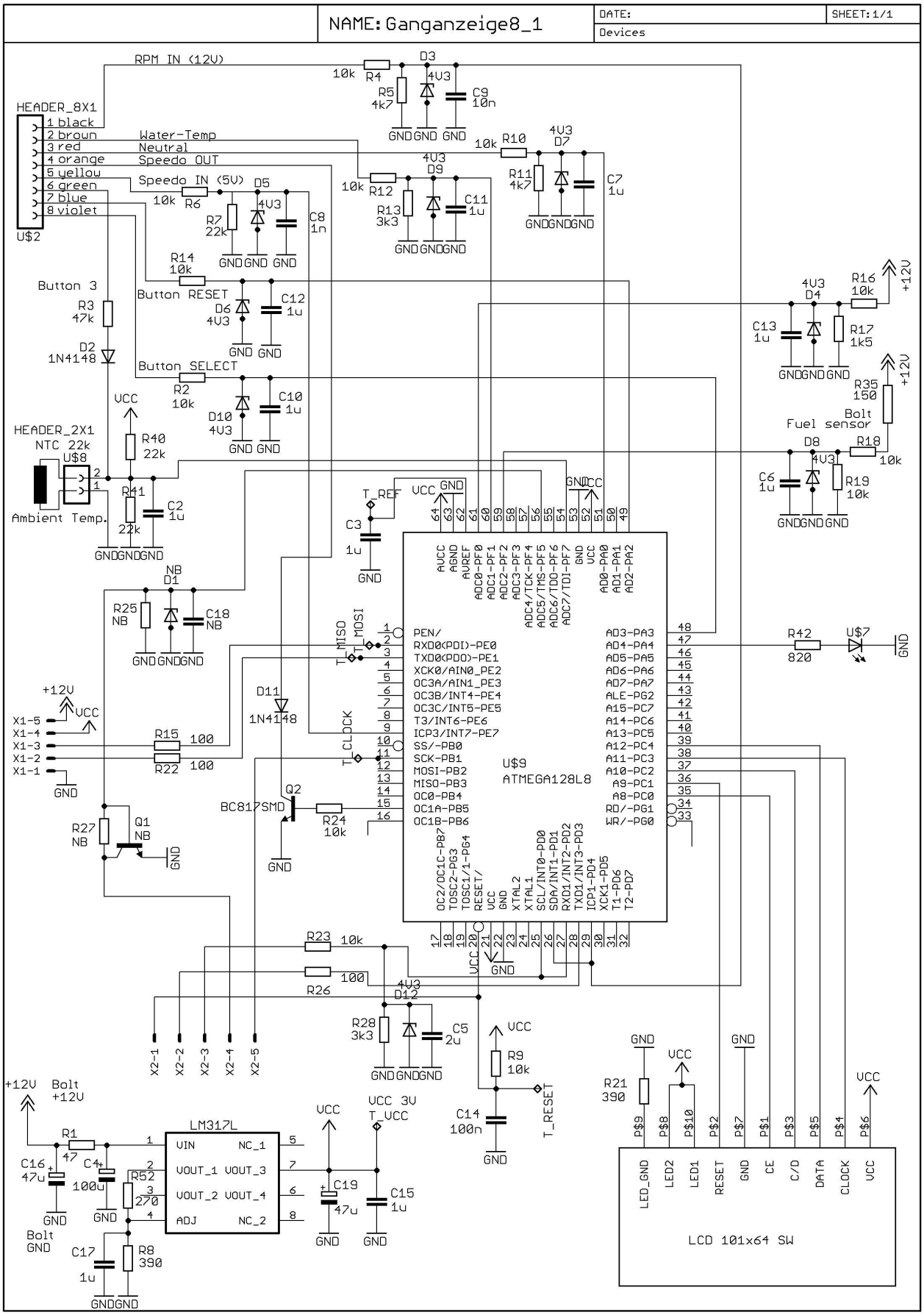




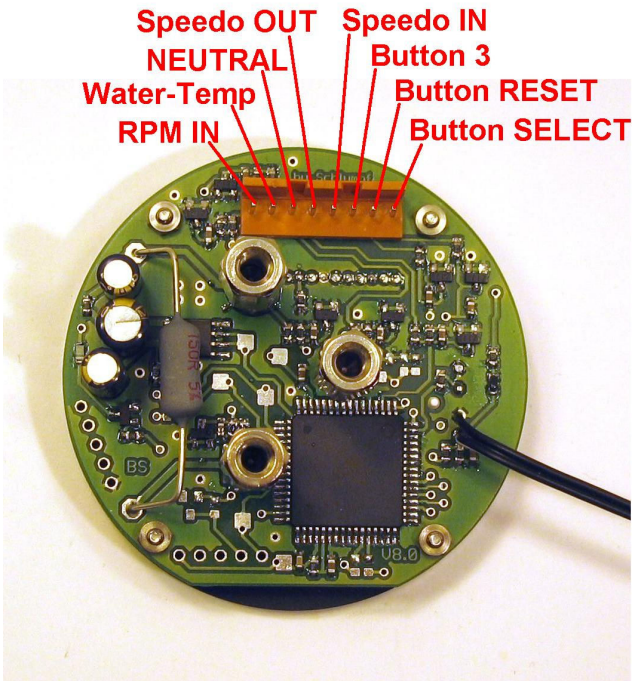


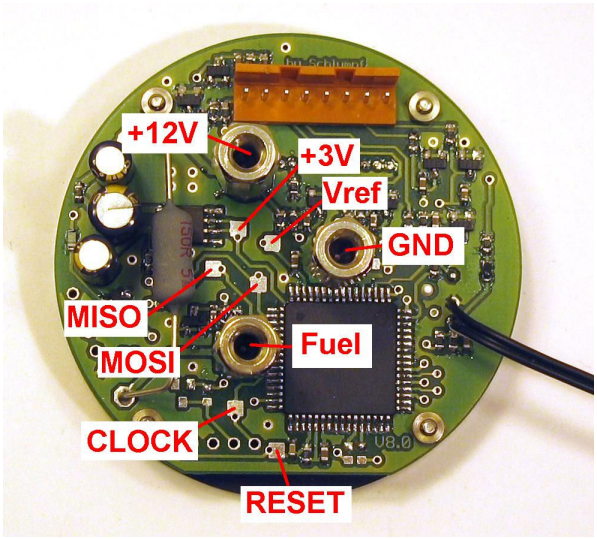
Multi Gauge schematic

Extension terminals: X1 and X2



Important signals





cvieth@arcor.de