

Die GPS - Tacho - Uhr im Saporoshez

Stand: 28.11.2021

Hallo zusammen,

wie viele Oldtimerfahrer auch schalte ich nach dem Abstellen meines Saporoshez 968 in der Garage die Batterie ab. Das Entladen der Batterie über Tage und Wochen wird damit vermieden bzw. stark verzögert. Die Folge ist jedoch, dass normale elektrische Uhren im Auto in dieser Zeit stehen bleiben.

Eine der möglichen Lösungen für dieses Problem ist eine Uhr, die sich automatisch nach dem GPS Satelliten einstellt. GPS Satelliten senden allerdings die [UTC-Zeit](#). Für eine Uhr in Mitteleuropa muss zu dieser UTC Zeit bei Winterzeit 1 Stunde, bei Sommerzeit 2 Stunden dazu addiert werden.

Mit Hilfe von GPS Satelliten kann man nicht nur die UTC-Zeit, sondern auch die relative Geschwindigkeit ermitteln, in meinem Fall in Kilometer pro Stunde. Es bietet sich somit an, mit den vorhandenen Bauteilen direkt eine kombinierte Uhr mit Tacho zu bauen.

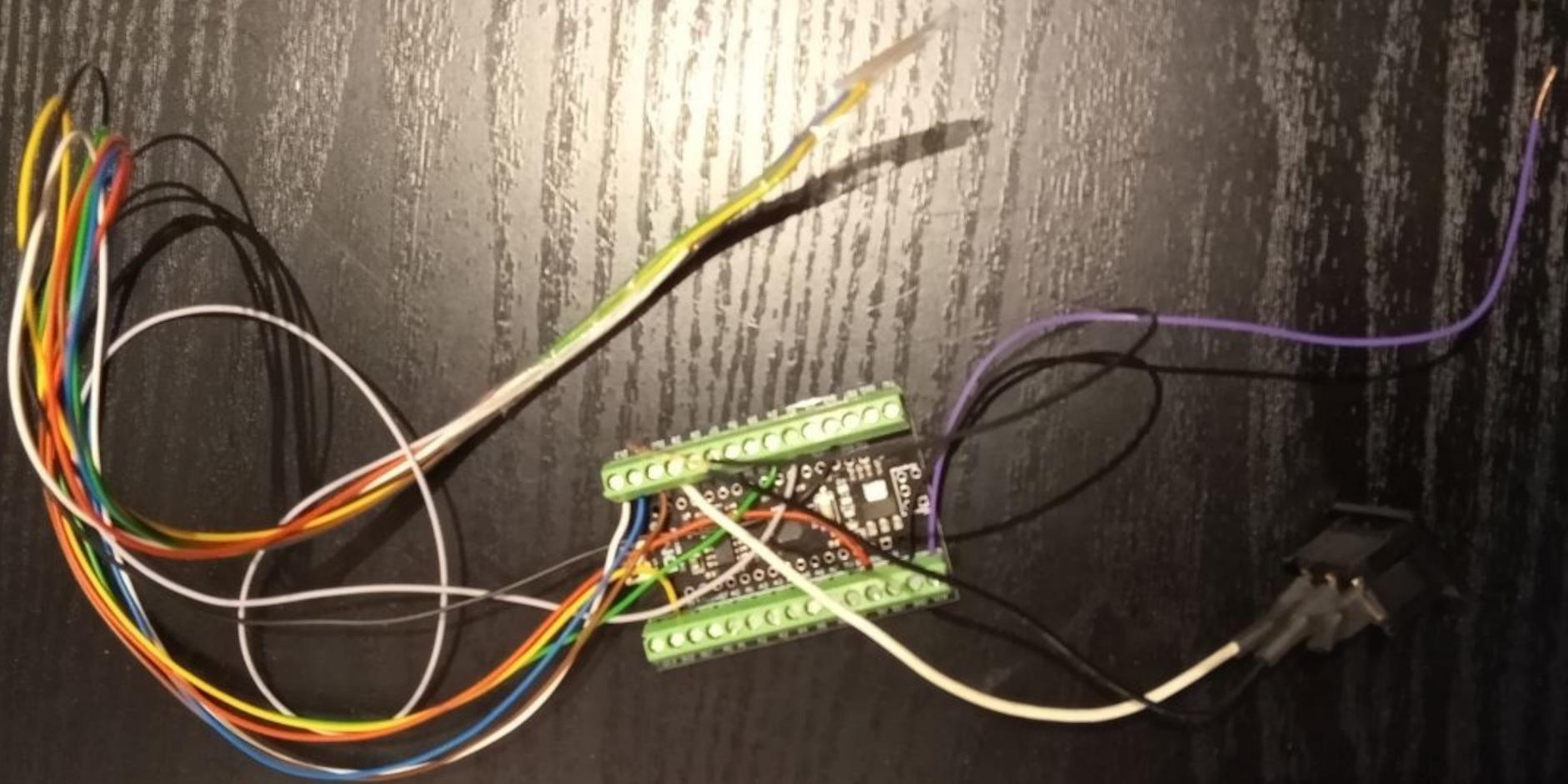
In der Praxis zeigt sich, dass nach etwa 20 Sekunden nach Verlassen der Tiefgarage eine Uhrzeit angezeigt wird, bis zur ersten Anzeige eines KMH-Signals vergehen etwa 3 Minuten.

Die Berechnungen übernimmt ein Arduino Nano, das GPS-Signal wird von einem GPS6MV2 Modul empfangen und die Ziffern werden von einer schön rot leuchtenden 7Segment LED-Anzeige mit Max7219 Ansteuerung angezeigt. Der Arduino wird von einem Spannungswandler Modul LM2596 zuverlässig mit 5 Volt versorgt. Diese 4 Bauteile kann man bei einem Onlinehändler in Deutschland für etwa Euro 20,00 bestellen.

Im Saporoshez wird der Arduino Nano zusammen mit dem LM2596 Modul im Armaturenbrett bzw. hinter der Regelung für die elektrische Choke-Betätigung versteckt. Das kleine Modul mit der GPS-Antenne findet noch Platz im Inneren des Drehzahlmessers, nun führen statt 4 Litzen insgesamt 8 Litzen aus dem Drehzahlmesser in das Armaturenbrett. Die 7Segment LED-Anzeige wird, unter einem schwarz lackierten Alu-Winkel geklebt, in den Spalt zwischen Instrumententräger und Armaturenbrett geklemmt.

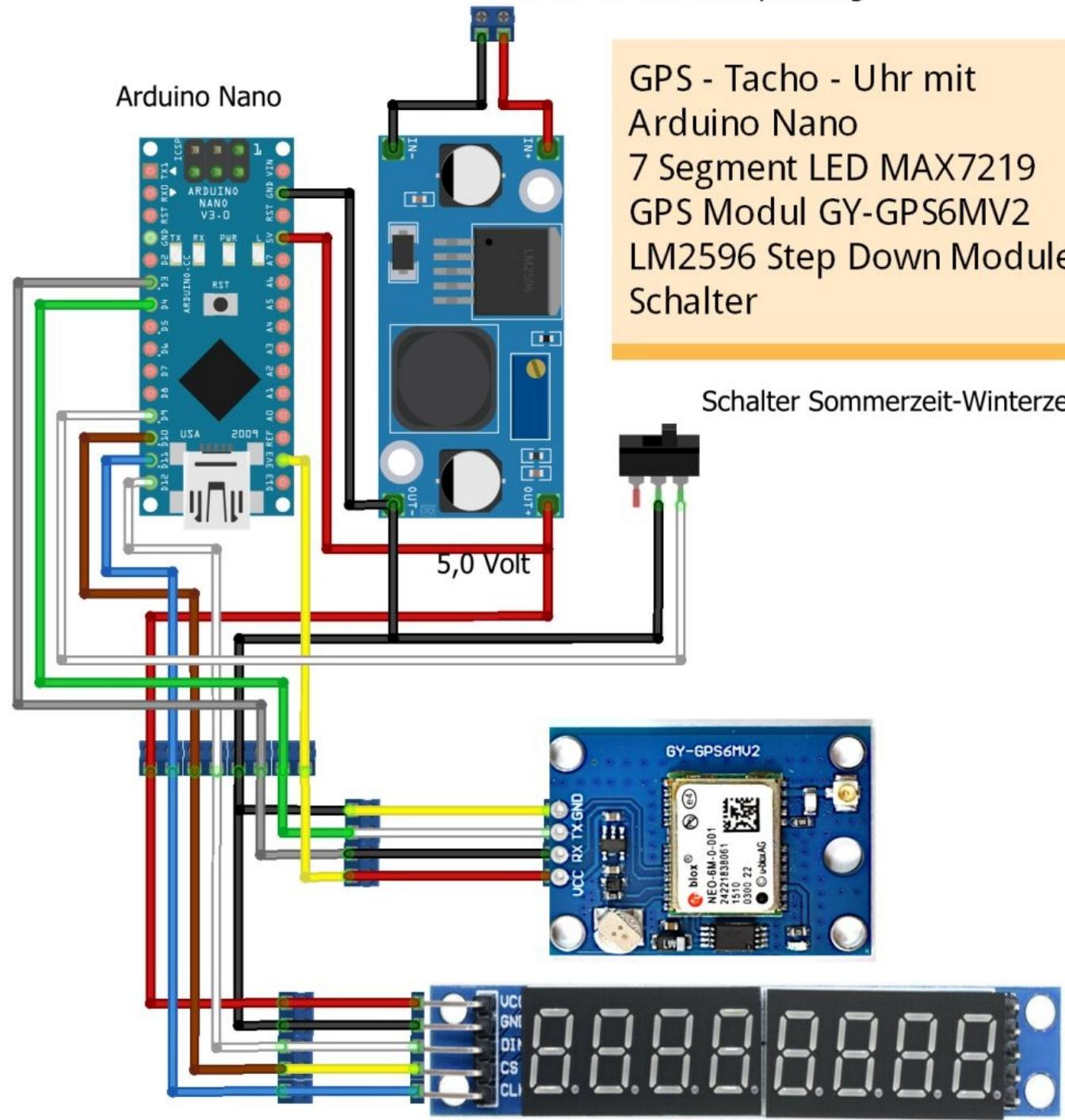
8888 15 19







10 bis 16 Volt Bordspannung



```

//gps_tacho_uhr_7-segment-led_sapo
// Nicht D2, soll freibleiben
// Nicht D5, soll freibleiben
//*****
// Anschlüsse 7Segment mit Max7219
// VCC - rotes Kabel - rotes Kabel - Arduino 5V
// GND - schwarzes Kabel - schwarzes Kabel - Arduino GND
// DIN - weißes Kabel - weißes Kabel - Arduino D12
// CS - gelbes Kabel - braunes Kabel - Arduino D10
// CLK - blaues Kabel - blaues Kabel - Arduino D11
//*****
// Anschlüsse GPS6MV2 Modul
// VCC - rotes Kabel - gelbes Kabel - Arduino 3,3V
// RX - schwarzes Kabel - graues Kabel - Arduino D3
// TX - weißes Kabel - grünes Kabel - Arduino D4
// GND - gelbes Kabel - schwarzes Kabel - Arduino GND
//*****
// Schalter für Sommerzeit
// weißes Kabel - Arduino D9
// schwarzes Kabel - Arduino GND
//*****
#include <SoftwareSerial.h>
#include <TinyGPSPPlus.h>
#include "LedControl.h"
int vers = 5;
int minutens = 0; // rohe Minuten
int hoursr = 0; // rohe Stunden eins
int hoursa = 0; // rohe Stunden zwei
int kmhgpsr = 0; // rohe Kilometer pro Stunde
int kmhgps = 0; // geglättete Kilometer pro Stunde
char c7 = '3'; // Das Zeichen an der Hunderter-Stelle
char c6 = 'A'; // Das Zeichen an der Zehner-Stelle
char c5 = '3'; // Das Zeichen an der Einer-Stelle
int z7 = 0 ; // Die Ziffer an der Hunderter-Stelle
int z6 = 0 ; // Die Ziffer an der Zehner-Stelle
int z5 = 0 ; // Die Ziffer an der Einer-Stelle
TinyGPSPPlus gps;
unsigned long previousMillis = 0; // will store last time LED was updated
const long interval = 800; // interval at which to blink (milliseconds)
int punktwei = 0;
SoftwareSerial ss(4, 3);
LedControl lc=LedControl(12,11,10,1);
void setup() {
  pinMode(9,INPUT_PULLUP);
  Serial.begin(9600);
  ss.begin(9600);
  /*

```

```

The MAX72XX is in power-saving mode on startup,
we have to do a wakeup call
*/
lc.shutdown(0, false);
/* Set the brightness to a medium values */
lc.setIntensity(0,0.1); // 8 ist hell, 0.1 ist abgedunkelt
/* and clear the display */
lc.clearDisplay(0);
}
void loop()
{
  if (digitalRead(9)==LOW)
  vers = 1 ; // mitteleuropäische Winterzeit MEZ ist UST+1
  else
  vers = 2 ; //mitteleuropäische Sommerzeit MESZ ist UST+2
  while (ss.available() > 0) {
  gps.encode(ss.read());
  if (gps.time.isUpdated()) {
  int kmhgpsr = gps.speed.kmph();
  // int kmhgpsr = 119;
  if ( kmhgpsr > 9 ) { kmhgps = kmhgpsr ; }
  else { kmhgps = 0 ; }
  int minutens = gps.time.minute();
  int hourasr = gps.time.hour() + vers;
  if ( hourasr < 24 ) { houras = hourasr ; }
  else { houras = hourasr - 24; } // Statt 24 Uhr ist es 00 Uhr
  int z7 = kmhgps / 100;
  int z6 = ( kmhgps - ( z7 * 100 ) ) / 10;
  int z5 = ( kmhgps - ( z7 * 100 ) - ( z6 * 10));
  if ( z7 > 0 ) { c7 = z7 ; }
  else { c7 = ' ' ; }
  if ( (z6+z7) > 0 ) { c6 = z6 ; }
  else { c6 = ' ' ; }
  if ( kmhgps > 0 ) { c5 = z5 ; }
  else { c5 = ' ' ; }
  if (gps.time.hour() < ( 10 + vers)) Serial.print(F("0"));
  Serial.print(gps.time.hour() + vers);
  Serial.print(F(":"));
  if (gps.time.minute() < 10) Serial.print(F("0"));
  Serial.println(gps.time.minute());
  Serial.println(minutens+3);
  lc.setDigit(0,0,(minutens - ((minutens / 10 ) * 10)),false);
  lc.setDigit(0,1,(minutens / 10 ),false);
  lc.setDigit(0,3,(houras / 10 ),false);
  lc.setChar(0,7,c7,false);
  lc.setChar(0,6,c6,false);
  lc.setChar(0,5,c5,false);

```

```
//Serial.println(F(":"));
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= interval) {
// save the last time you blinked the LED
previousMillis = currentMillis;
// if the LED is off turn it on and vice-versa:
if (punktzwei == 0)
{
lc.setDigit(0,2,(hours - ((hours / 10 ) * 10)),false);
punktzwei = 1;
} else {
lc.setDigit(0,2,(hours - ((hours / 10 ) * 10)),true);
punktzwei = 0;
}
}
}
}
}
}
```

[zurück zur Hauptseite](#)
www.saporoshez-968.de