



GPS-gebaseerde snelheidsmeter

nooit meer bekeuringen

Olivier Croiset (Frankrijk)

Er zijn veel voorbeelden op internet die laten zien hoe je een GPS-module kunt gebruiken in een microcontroller-project. Maar wat kun je met die GPS-coördinaten doen, behalve ze op een kaart in te tekenen? Ik wilde iets bruikbaars maken en bedacht deze GPS-gebaseerde snelheidsmeter. Dit apparaat kan in elk voertuig worden gebruikt en stelt je in staat te rijden zonder op de snelheidsmeter te hoeven letten, en voorkomt onaangename verrassingen achteraf in de brievenbus.

Het hier gepresenteerde apparaat is geen snelheidsbegrenzer, waarvoor aanpassing aan het voertuig nodig zou zijn, maar een alarm dat signaleert wanneer een vooraf ingestelde snelheid wordt bereikt (en overschreden). Om ervoor te zorgen dat je je ogen op de weg

gericht kan houden, laat de zoemer twee lange pieptonen horen als de ingestelde snelheid wordt overschreden, terwijl twee korte pieptonen aangeven dat het voertuig weer een snelheid onder de grenswaarde heeft bereikt.

Vier presets

De snelheidsmonitor heeft vier programmeerbare snelheidslimieten, wat voldoende zou moeten zijn voor de meeste ritten en situaties die je tegenkomt. Bovendien laat het scherm nauwelijks meer dan die vier limieten zien. Onder het rijden moet de bestuurder snel een preset kunnen kiezen, indien mogelijk zonder naar het apparaat te kijken. Ik wilde geen aanraakscherm gebruiken; de bestuurder moet de knopdruk fysiek ervaren. Om ergonomische redenen zijn de knoppen boven de vier snelheidswaarden op het display geplaatst.

Een goed excuus om met 32 bit aan de slag te gaan

Mijn eerste programma's met een 8-bit ATmega328-microcontroller in combinatie met een GPS-module bewezen al snel dat het 32 KB flash-geheugen niet genoeg zou zijn voor een meer geavanceerde toepassing, vooral als de gegevens grafisch op een TFT-scherm moeten worden weergegeven.

Daarom besloot ik, hoewel overtuigd van de vele kwaliteiten van de ATmega328, het hogerop te zoeken en



een STM32 32bit-microcontroller uit te proberen. De meest geschikte uitvoering voor knutselaars zoals ik is het BluePill-board met een STM32F103C8-chip. Dit kleine board vergemakkelijkt het gebruik van deze microcontroller aanzienlijk. Het is eenvoudig te programmeren via de USB-poort met behulp van de Arduino IDE (mits het board vooraf is voorzien van een geschikte bootloader), het is compact en niet duur (slechts een paar euro).

Flash-geheugenformaten

Dit project is mijn eerste poging om de STM32F103C8 te gebruiken. Merk op dat officieel de C8 is uitgerust met 64 KB flash, terwijl de CB-versie 128 KB bezit, maar soms komt de C8 ook met 128 KB (namaak?). Hierdoor kunnen degenen die in de verleiding komen, hun kennis vergroten door andere periferie toe te voegen, zoals een SD-kaart, een SIM-module, draadloze gegevensoverdracht enzovoort.

De GPS-module

De GPS-module die in dit project wordt gebruikt, is een NEO-6-module van u-blox. Het gebruik ervan is eenvoudig, omdat de TinyGPS+ bibliotheek [2] al het decodeerwerk van de NMEA 0183-geformatteerde gegevens-

stroom die door de GPS-module wordt geleverd, voor zijn rekening neemt. Dit zijn de belangrijkste parameters die we op deze manier verkrijgen:

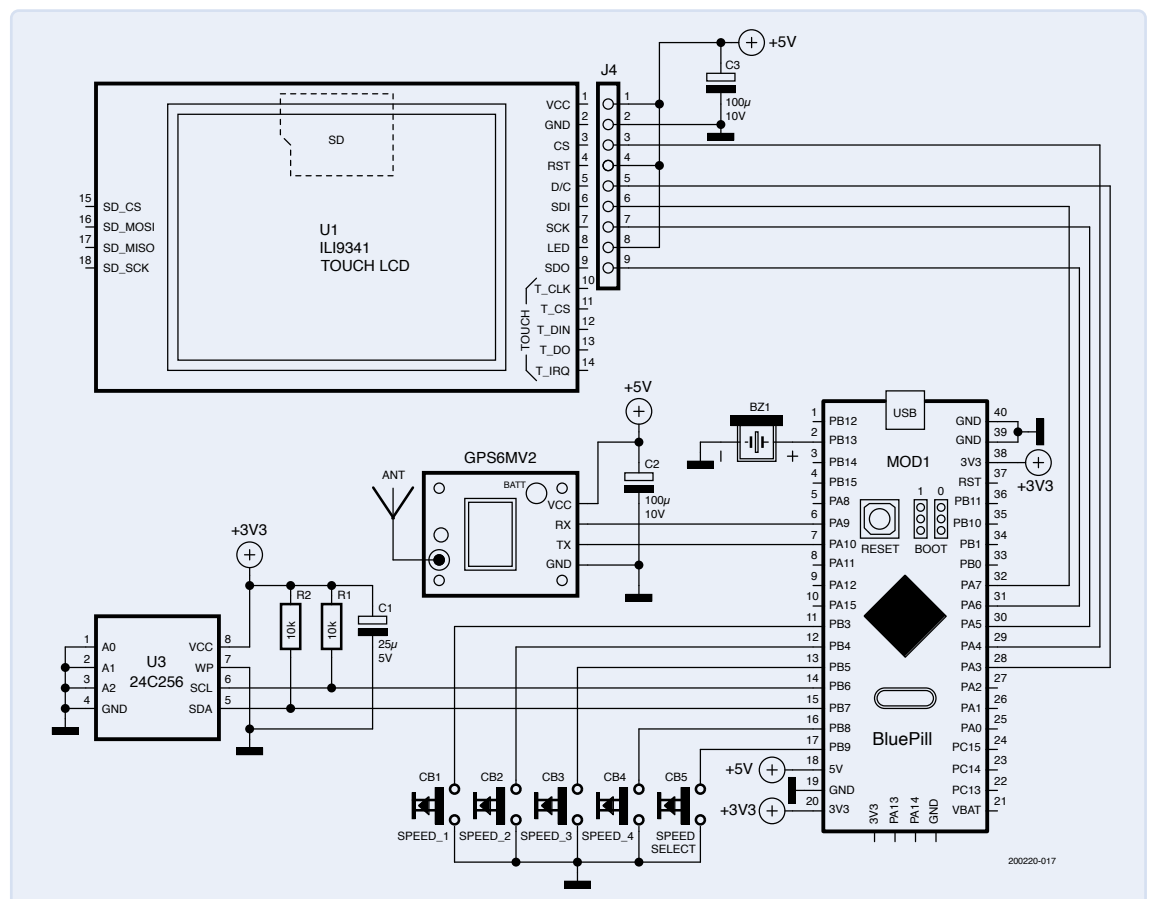
- UTC-datum en -tijd;
- lengte- en breedtegraad (in decimale graden);
- snelheid;
- richting;
- hoogte;
- het aantal zichtbare satellieten;
- de HDOP-waarde.

De *Horizontal Dilution of Precision* ofwel HDOP [2] hangt af van de positie van de satellieten binnen het bereik van de GPS-ontvanger. Hoe lager deze waarde is (dichtbij 1), des te nauwkeuriger de berekende coördinaten. Een HDOP-waarde van 10 geeft aan dat de coördinaten niet erg nauwkeurig of zelfs ongeldig zijn. De snelheidsmonitor gebruikt deze parameter niet.

Het TFT-scherm

Als display gebruikte ik een standaard 2,2" TFT-display-module met een ILI9341-driverchip en SPI-interface. Het heeft een resolutie van 320×240 pixels, wat genoeg is voor de gegevens die we willen weergeven (vier

Figuur 1. De snelheidsmonitor combineert kant-en-klare modules met een paar elektronische componenten.



snelheidslimieten plus een kort bericht dat aangeeft of de limiet is bereikt of niet).

De snelheidsmonitor heeft bij normaal gebruik twee hoofd-weergavemodi. De momentele weergavemodus wordt opgeslagen in het EEPROM en wordt opgeroepen wanneer het systeem opnieuw wordt ingeschakeld. De snelheids-presets worden ook opgeslagen in EEPROM. Van het EEPROM worden slechts zes bytes gebruikt: vier bytes voor de presets, één byte voor het laatst geselecteerde scherm en één byte voor de laatst geselecteerde snelheidslimiet.

Het schema

Het schema van de snelheidsmonitor is gegeven in **figuur 1**. De BluePill-module in de rechterbenedenhoek vormt het hart van de schakeling.

De voeding wordt aangesloten via de USB-connector van de BluePill. Nieuwere voertuigen zijn vaak voorzien van een USB-aansluiting; anders is een 12V-naar-USB-adapter vereist (of gebruik een powerbank). De BluePill-module wordt gevoed met 5 V; hij heeft zijn eigen 3,3V-regelaar. Deze laatste voorziet het EEPROM van 3,3 V. Het TFT-display en de GPS-module worden gevoed met 5 V. Het totale verbruik van de schakeling is ongeveer 200 mA. Het display wordt aangestuurd via een SPI-bus, en de GPS-module wordt aangestuurd via een seriële link. Het EEPROM is aangesloten op de I²C-bus (met zijn twee pull-up weerstanden). Voor mijn prototype gebruikte ik een 32 KB 24C256 EEPROM, maar een 24C01 van 128 byte zou al meer dan groot genoeg zijn.

Twaalf GPIO-poorten blijven vrij. Het is dus mogelijk om meer periferie, features en functionaliteit toe te voegen – zolang het applicatieprogramma maar in het flash-geheugen past. De gecompileerde firmware van dit project neemt 64.020 bytes in beslag en daarmee bijna het volledige flash-geheugen.

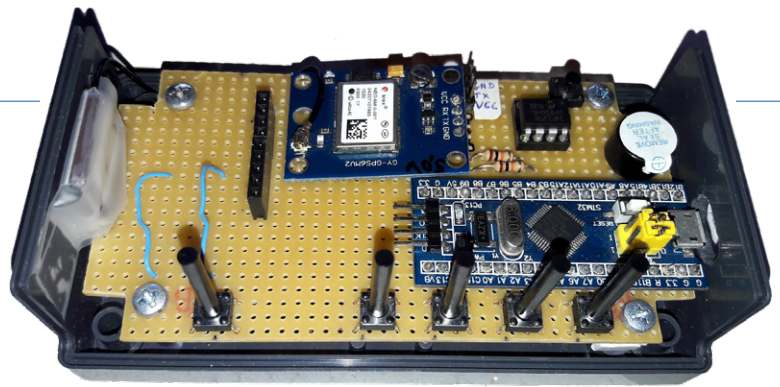
Bouw van de snelheidsmonitor

Het belangrijkste criterium bij de bouw van dit project is ergonomie. De chauffeur moet de knoppen blindelings kunnen vinden. De druktoetsen hebben een toetslengte van 25 mm – de langste die ik kon vinden.

Ik heb mijn prototype opgebouwd op gaatjesprint (**figuur 2**). Een echte print (PCB) zou een compactere uitvoering mogelijk maken in een plattere behuizing. De USB-connector moet toegankelijk blijven, aangezien deze wordt gebruikt om het circuit bij normaal gebruik van stroom te voorzien en om de software te actualiseren. De GPS-antenne moet zo ver mogelijk van de TFT-displayconnector verwijderd blijven (minstens enkele centimeters).

Enkele opmerkingen over de GPS-module

De GPS-module heeft een kleine back-up batterij voor data. Als de GPS een paar dagen achter elkaar niet wordt



gebruikt, raakt de batterij leeg en gaan de gegevens verloren. Wanneer je de GPS weer inschakelt, kan het zijn dat je enkele minuten moet wachten om de gegevens te herstellen (en de batterij op te laden). De LED van de GPS-module knippert zodra de satellietgegevens met succes zijn gedecodeerd. Even later wordt het aantal satellieten weergegeven.

De GPS-antenne moet de satellieten 'zien'. De ontvangst van de signalen moet daarom zo min mogelijk worden belemmerd. In stedelijke gebieden, tussen twee gebouwen, kan de ontvangst moeilijk zijn. Desalniettemin werkt de snelheidsmeter in mijn auto naar behoren, ook als die in de voetruimte van de passagier is geplaatst.

Voorbereiding van de snelheidslimieten

Dit mag alleen worden gedaan als het voertuig stilstaat! Bij de eerste keer opstarten van de snelheidsmonitor zijn de snelheidslimieten allemaal ingesteld op 255. Dat komt overeen met FF hexadecimaal, en dat is wat er in een lege EEPROM staat.

Figuur 2: Het prototype op gaatjesprint. De afmetingen van de behuizing zijn 12 x 6,5 x 4 cm.

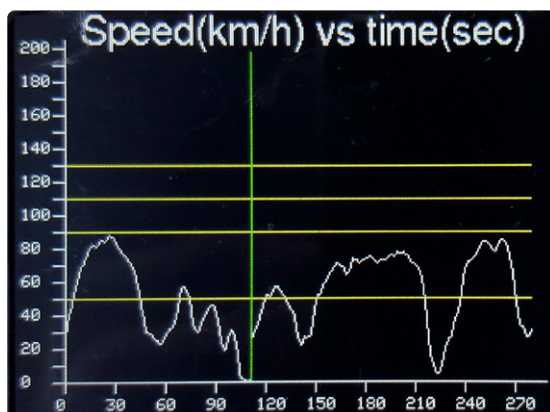


Figuur 3: Scherm 1 toont de snelheidslimieten als ronde verkeersborden met een korte tekst.



Figuur 4: In scherm 2 wordt de werkelijke snelheid weergegeven via een gesimuleerd 7-segment display.

Figuur 5. De curve van de snelheidsgeschiedenis wordt weergegeven in scherm 5. De vier gele horizontale lijnen geven de vier geprogrammeerde limieten weer en de groene verticale lijn geeft de huidige snelheid weer.



In scherm 3 (figuur 5), dat de gegevens van de satellieten toont, druk je op de derde knop om de instelling van de snelheidslimieten te activeren. Gebruik vervolgens de eerste en tweede knop om de gewenste limiet in te stellen. Druk nogmaals op de derde knop om de snelheidslimiet te valideren en door te gaan naar de volgende – en zo vervolgens. Zorg ervoor dat de limieten in een logische volgorde staan, aangezien de software ze niet sorteert.

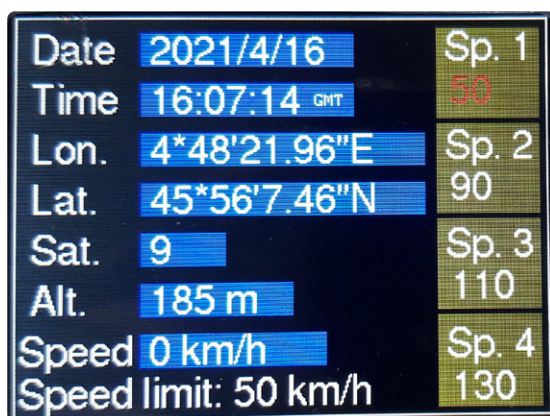
Gebruik van de snelheidsmonitor

Het gebruik van de snelheidsmonitor is niet moeilijk. Aangezien hij bedoeld is voor gebruik in een rijdend voertuig, moet je ervoor zorgen dat het apparaat en vooral de voedingskabel niet in de weg zit tijdens het rijden. Zodra de monitor in een geschikte positie is geplaatst, selecteer je de gewenste weergavemodus. Scherm 1 (figuur 3) toont de bekende ronde verkeersborden, scherm 2 geeft de momentane snelheid weer met 7-segment cijfers (figuur 4). Selecteer vervolgens de snelheidslimiet die je wilt gebruiken. De zoemer klinkt als je te snel gaat. De passagier kan de snelheidscurve over een periode van 4,5 minuut (270 seconden) bekijken door scherm 5 te selecteren (figuur 5). In deze modus klinkt de zoemer niet. De horizontale lijnen geven de vier maximumsnelheden weer. De verticale lijn die 'nu' aangeeft, toont de huidige snelheid. Een scrollende grafiek was misschien mooier geweest, maar daarvoor moet het scherm voor elk nieuw gegevenspunt compleet worden ververs, en dat is traag. Het verplaatsen van de cursor is veel eenvoudiger.

Figuur 6. GPS-gegevens zijn te zien in scherm 3. Dit is ook het scherm dat toegang geeft tot de snelheidslimiet-presets in scherm 4.



Figuur 7. In scherm 4 kun je de snelheidslimieten instellen. Hier wordt de eerste grenswaarde ('Sp. 1') ingesteld (in het rood, als je goed kijkt).



GPS-gegevensscherm

Scherf 3 (figuur 6) toont de GPS-gegevens en stelt je in staat je locatie te vinden op een kaart waarop coördinaten zijn aangegeven. De nauwkeurigheid is ongeveer 30 meter, wat best goed is voor ons kleine en goedkope apparaatje. In deze weergavemodus klinkt de zoemer niet. Scherm 4 (figuur 7) is hetzelfde als scherm 3, behalve dat je hier de snelheidslimieten kunt instellen.

Tabel 1. De verschillende schermen en hoe je er doorheen navigeert.

	Knop 1	Knop 2	Knop 3	Knop 4	Knop 5
Scherf 1 (ronde verkeersborden)	Limiet 1	Limiet 2	Limiet 3	Limiet 4	Ga naar scherm 2
Scherf 2 (7-segment display)	Limiet 1	Limiet 2	Limiet 3	Limiet 4	Ga naar scherm 3
Scherf 3 (GPS-gegevens)	Limiet 1	Limiet 2	Ga naar scherm 4	Geen functie	Ga naar scherm 1
Scherf 4 (stel limieten in)	Verlaag limiet (-5 km/u)	Verhoog limiet (+5 km/u)	Bevestig en naar volgende limiet, terug naar scherm 3 als je klaar bent	Ga naar scherm 5	Geen functie
Scherf 5 (snelheidscurve)	Geen functie	Geen functie	Geen functie	Ga naar scherm 2	Geen functie

Merk op dat de GPS-datum en -tijd UTC-waarden zijn, dus betrokken op de meridiaan van Greenwich. Als je ze wilt gebruiken, moet je deze waarden omrekenen naar je lokale tijdzone (en eventueel zomer- en wintertijd). Aangezien de snelheidsmonitor geen klok is, is hiervoor niet in een comfortabel menu voorzien.

De software

Het programma kan eenvoudig worden aangepast; de broncode kan worden gedownload van [3]. Het is een Arduino-sketch die het STM32 Boards-package voor Arduino nodig heeft (wij gebruikten v2.3.0) [4]. Het leeuwendeel van de code betreft de grafische gebruikersinterface en de rest zorgt voor de eenvoudige taak van het ontvangen van GPS-gegevens, het extraheren van de snelheid en het vergelijken met een grenswaarde. Zoals gebruikelijk in een Arduino-sketch zorgt de functie `setup()` voor het initialiseren van alle periferie. Als dat gebeurd is, wordt gedurende vijf seconden een welkomstscherm weergegeven.

De `loop()`-functie begint met het inlezen van de drukknoppen voordat de GPS wordt gecontroleerd op nieuwe gegevens (in de `dataDecode()`-functie). Vervolgens worden, afhankelijk van welk scherm actief is, de bijbehorende gegevens weergegeven. Als de momentele snelheid hoger is dan de geselecteerde snelheidslimiet, klinkt een alarm.

Merk op dat debug- en statusinformatie wordt verzonden via de seriële poort (115200,N,8,1).

Aanpassingen

Hier zijn een paar dingen die je misschien wilt toevoegen of veranderen:

- Andere eenheden voor de snelheid. Met de GPS--bibliotheek kun je andere eenheden gebruiken, zoals knopen of mijlen/uur (mph). Hiervoor moeten ook de schermen worden aangepast (bijvoorbeeld km/u vervangen door knopen) en de stapgrootte voor het verhogen van de limieten (in de `SpeedSettings()`-functie). Dit kan handig zijn voor een boot.

Als je BluePill-module 128 KB flash bezit, kunnen extra functies worden toegevoegd zoals:

- Opname op SD-kaart. Aan de achterkant van de TFT-displaymodule bevindt zich een SD-kaartsleuf. Deze kan worden gebruikt om bijvoorbeeld de GPS-gegevens op te slaan voor later gebruik op een PC.
- Voeg een aanraakscherm toe voor nieuwe functies. Zorg in dat geval voor de veiligheid van de gebruiker.
- Mogelijkheid om de lokale tijd in te stellen. ◀

200220-03

Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de redactie van Elektor via redactie@elektor.com.



Gerelateerde producten

- **OPEN-SMART GPS – Serial GPS Module for Arduino (SKU 18733)**
www.elektor.nl/18733
- **2.2" SPI TFT Display Module ILI9341 (240x320) (SKU 18419)**
www.elektor.nl/18419
- **Majid Pakdel, Advanced Programming with STM32 Microcontrollers (SKU 19520)**
www.elektor.nl/19520

WEBLINKS

- [1] TinyGPS++ bibliotheek: <http://arduinoiana.org/libraries/tinygpsplus/>
- [2] What exactly is HDOP?: [https://en.wikipedia.org/wiki/Dilution_of_precision_\(navigation\)](https://en.wikipedia.org/wiki/Dilution_of_precision_(navigation))
- [3] Projectbestanden op Elektor Labs:
<https://elektormagazine.com/labs/save-money-with-this-speed-monitoring-by-gps>
- [4] STM32 Boards-package voor Arduino:
https://github.com/stm32duino/BoardManagerFiles/raw/main/package_stmicroelectronics_index.json