Arduino en GPS-data
(deel 1)

door ON4CDU

Met een Arduino is het heel eenvoudig om GPS-data te bekijken. Het GPS-signaal kun je ook gebruiken om een nauwkeurige frequentiestandaard te maken.

Inleiding

Arduino en GPS-dataontvangst bieden veel uitdagingen om daar eens mee te experimenteren. Dit artikel beschrijft hoe je een 'GPS Display Unit' kunt maken en GPS voor frequentiestabilisatie kunt gebruiken.
Maar is daar dan geen appje voor? Jazeker, voor de GPS-data bestaat dat. Op je smartphone kun je een app installeren om de belangrijkste data zichtbaar te maken, met de locatie-indicatie desgewenst ook in de Maidenhead-notatie. Maar, met een variant op het oude gezegde 'het geluid uit een zelfgebouwde ontvanger klinkt beter': de gegevens op je zelfgebouwde GPS-display zijn duidelijker…
Het GPS (Global Positioning System) heeft de unieke eigenschap dat je het zowel voor plaatsbepaling als voor tijd (= frequentie) referentie kunt gebruiken. Een aantal mogelijkheden voor het praktisch gebruik door radio-amateurs komt in dit artikel aan bod. Om de experimenten met de Arduino te kunnen uitvoeren, wordt een kleine voorkennis verondersteld, namelijk dat de voorbeelden uit de Arduino IDE geen probleem opleveren, dat men een nieuwe bibliotheek kan laden, en dat men de seriële monitor begrijpt. Wie die basiskennis nog niet heeft, kan op internet hierover veel stuff vinden; bijvoorbeeld het Arduino Cookbook, of blader eens door de workshoppapieren op mijn internetsite: http://on4cdu.net/arduino-workshop/.

GPS-data

GPS-ontvangers leveren data in NMEA-formaat. Het protocol bestaat uit losse berichten waarin gegevens gegroepeerd zijn. De berichten bestaan uit leesbare ASCII-karakters plus de 'carriage return' en 'line feed' karakters. Het NMEA-protocol is op het internet goed beschreven. Bekijken we GPS NMEA-data op een display, dan zien we iets als in afbeelding 1.
De berichten beginnen met een '$' en een identificatiewoord. Daarna volgt de data, gescheiden door komma's, en aan het einde van het bericht een checksum.
Als voorbeeld het bericht met het identificatiewoord GPRMC (GPS Recommended Minimum data): $GPRMC,115452.00,A,5046.93931,N,00431.96130,E,0.10,,060715,,,D*7A. De data tussen de komma's heeft de volgende betekenis:

```
$GPGSV,4,3,13,26,07,278,,29,66,203,44,31,43,303,42,33,29,205,34*70
$GPGSV,4,4,13,39,29,154,47*49
$GPGLL,5046.93922,N,00431.96128,E,115450.00,A,D*6E
$GPRMC,115451.00,A,5046.93926,N,00431.96130,E,0.020,,060715,,,D*7C
$GPVTG,,T,,M,0.020,N,0.036,K,D*21
$GPGGA,115451.00,5046.93926,N,00431.96130,E,2,07,1.16,107.5,M,46.2,M,,0000*53
$GPGSA,A,3,02,29,12,25,31,24,06,,,,,1.82,1.16,1.39*0B
$GPGSV,4,1,13,02,34,066,51,06,11,031,44,10,00,050,,12,39,087,50*7C
$GPGSV,4,2,13,14,24,232,,21,00,179,,24,07,148,45,25,79,061,45*73
$GPGSV,4,3,13,26,07,278,,29,66,203,44,31,43,303,42,33,29,205,34*70
$GPGSV,4,4,13,39,29,154,47*49
$GPGLL,5046.93926,N,00431.96130,E,115451.00,A,D*62
$GPRMC,115452.00,A,5046.93931,N,00431.96130,E,0.010,,060715,,,D*7A
$GPVTG,,T,,M,0.010,N,0.019,K,D*2F
$GPGGA,115452.00,5046.93931,N,00431.96130,E,2,07,1.16,107.3,M,46.2,M,,0000*50
$GPGSA,A,3,02,29,12,25,31,24,06,,,,,1.82,1.16,1.39*0B
```

Afb. 1. GPS-data */ Fig. 1. Données GPS*

Bericht	Voorbeeld	Beschrijving/notatie
Berichtnaam	$GPRMC	RMC bericht
Tijd in UTC	115452.00	hhmmss.ss
Geldigheid van het bericht	A	A (Active = OK) of V (Void = NOK)
Breedtegraad	5046.93931	ddmm.mmmmm
Halfrond	N	noord (N) of zuid (S) van de evenaar
Lengtegraad	00431.96130	dddmm,mmmm
Lengte oost/west	E	oost (E) of west (W) van de nulmeridiaan
Snelheid t.o.v. grond	0.10	in knopen (1 knoop = 1,852 km/h)
Ware koers	-	graden t.o.v. ware noorden
Datum	060715	ddmmyy
Magnetische variatie	-	in graden; niet altijd aanwezig
Richting magnetische variatie	-	oost (E) of west (W); niet altijd aanwezig
Checksum	7A	

Arduino et données GPS
(1ère partie)

par ON4CDU – traduit par ON5WF

Avec un Arduino, il est très simple d'examiner des données GPS. Le signal GPS peut aussi être utilisé pour réaliser un étalon de fréquence précis.

Introduction

L'Arduino et la réception des données GPS offrent beaucoup de possibilités d'expérimentation. L'objet de cet article est d'expliquer comment réaliser une unité d'affichage GPS et utiliser le GPS pour obtenir un étalon de fréquence. Bien sûr, il existe une application permettant de visualiser, sur un smartphone, les données GPS les plus importantes, avec, si on le désire, les données de localisation en notation Maidenhead. Cependant, de même que selon le vieux dicton "le son du récepteur que l'on a construit soi-même est meilleur", les données de l'affichage GPS de construction personnelle sont plus claires.
Le GPS (Global Positioning System) présente la particularité de pouvoir être utilisé aussi bien pour la localisation que comme référence de temps (= fréquence). Cet article présente un certain nombre de possibilités pour son usage pratique par des radioamateurs. La mise en œuvre des expérimentations avec l'Arduino suppose un minimum de connaissances de base ; notamment que les exemples de l'Arduino IDE ne posent pas de problème, que l'on est capable de charger une nouvelle bibliothèque et que l'on comprend le principe du moniteur série. Si l'on ne possède pas ces connaissances de base, on peut trouver sur internet de quoi combler ces lacunes, comme par exemple l'Arduino Cookbook. On peu aussi consulter mon site : http://on4cdu.net/arduino-workshop/.

Données GPS

Les récepteurs GPS fournissent les données dans le format NMEA. Le protocole consiste en messages détachés dans lesquels les données sont groupées. Les messages se composent de caractères ASCII lisibles plus les caractères 'carriage return' et 'line feed'. Le protocole NMEA est bien décrit sur internet. La figure 1 montre ce que nous voyons en observant les données GPS NMEA sur un afficheur.
Les données commencent avec un '$' et un mot d'identification. Ensuite viennent les données, séparées par des virgules, et à la fin du message, une somme de contrôle.
Considérons par exemple le message avec le mot d'identification GPRMC (GPS Recommended Minimum data) : $GPRMC,115452.00,A,5046.93931,N,00431.96130,E,0.10,,060715,,,D*7A. Les données entre les virgules ont les significations suivantes :

Messages	Exemple	Description/note
Nom du message	$GPRMC	message RMC
Temps UTC	115452.00	hhmmss.ss
Validité du message	A	A (Active = OK) of V (Void = NOK)
Latitude	5046.93931	ddmm.mmmmm
Hémisphère	N	nord (N) ou sud (S)
Longitude	00431.96130	dddmm,mmmm
Position par rapport au méridien de Greenwich	E	à l'est (E) ou à l'ouest (W) du méridien de Greenwich
Vitesse par rapport au sol	0.10	en noeuds (1 noeud = 1,852 km/h)
Cap vrai	-	en degrés par rapport au pôle nord vrai
Date	060715	ddmmyy
Déclinaison magnétique	-	en degrés ; pas toujours présent
Dir. déclinaison magnétique	-	est (E) ou ouest (W) ; pas toujours présent
Somme de contrôle	7A	

De voor ons meest interessante data zijn opgenomen in de GPRMC en GPGGA strings. Datum, tijd, positie en hoogte zijn voor de hand liggende gegevens. Daarnaast willen we graag iets weten over de betrouwbaarheid van deze informatie. Hiervoor zijn de geldigheid van het bericht (A of V), data uit de GPGGA string (FIX = 0: onvoldoende informatie voor betrouwbare berekening, FIX >1: berekening betrouwbaar) en de HDOP (Horizontal Dilution Of Precision) kandidaten. Als extra is ook het aantal satellieten waarvan de gegevens verwerkt worden in de ontvanger een goede bron. Let ook eens op de notatie van de positiegegevens in graden en decimale minuten. Het blijkt dat het aantal cijfers achter het decimaalteken niet bij alle GPS-ontvangers hetzelfde is.

Arduino GPS-display

Interessant en leerzaam is het om de GPS-ontvanger aan een Arduino Uno of Nano aan te sluiten en de datastrings op een computerscherm te bekijken. Er bestaan veel GPS-ontvangers die geschikt zijn om op een Arduino aan te sluiten. Erg populair en goedkoop op internet zijn printjes met de aanduiding GY-NEO6MV2. Deze printjes, met een u-blox IC erop, worden met een actieve patchantenne geleverd. De NMEA-berichten zijn beschikbaar op de TX-uitgang van het printje, en de standaardsnelheid daarvan is 9600 bits/sec.

Een GY-NEO6MV2 module met patchantenne

Un module GY-NEO6MV2 avec antenne patch

De 'oude' Rockwell Jupiter GPS-ontvanger is ook te gebruiken. Deze ontvangers zijn/waren zeer gewild bij radioamateurs, omdat ze verkrijgbaar en goedkoop waren, en ook nog een 10 kHz uitgangssignaal afgeven dat voor synchronisatiedoeleinden gebruikt kan worden. De aansluitingen van de Rockwell Jupiter staan goed beschreven op http://www.gpskit.nl/. De standaard bitsnelheid van de NMEA-uitgang is hier 4800 bits/sec.

Rockwell Jupiter GPS-ontvanger

Récepteur GPS Rockwell Jupiter

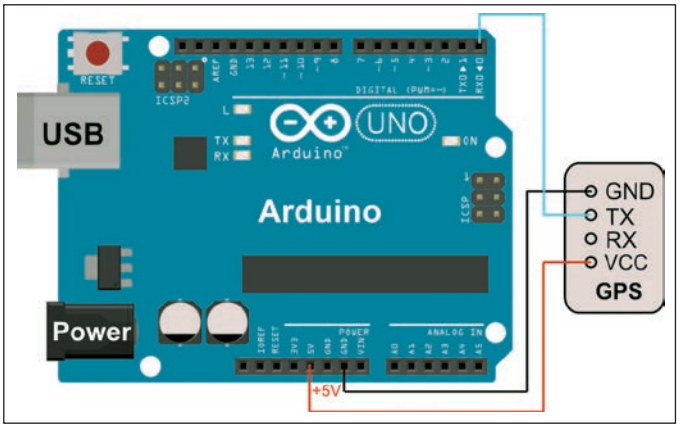
Start de Arduino IDE op de pc en laad hierin het programma 'GPS-string naar serial monitor'. Deze software is te downloaden van: http://on4cdu.net/arduino-en-gps-data/ en dan: Arduino GPS software. Stel de datasnelheid voor de GPS-ontvanger in (Jupiter 4800 en u-blox 9600 bits/sec) en laad het programma in de Arduino. Sluit nu de NMEA-uitgang van de ontvanger aan op de ingang van de UART van de Arduino (pin RX bij Uno; RX0 bij Nano). Met de seriële monitor zal nu de GPS-data op het scherm verschijnen. Denk eraan de seriële monitor op de juiste datasnelheid in te stellen, en wanneer een programma in de Arduino geladen moet worden, de ontvanger los te koppelen van pin RX van de Arduino; deze input wordt ook gebruikt om nieuwe software te laden!
Als alles goed gegaan is, zijn de GPS-strings zichtbaar op het scherm. Het is interessant de data te bestuderen en te interpreteren. Trek ook de GPS-antenne eens los en kijk wat er dan gebeurt.
Moderne ontvangers starten veel sneller op en reageren ook sneller op veranderende omstandigheden dan de oude Jupiters. Ze zijn daarom ook geschikt voor 'flight control' toepassingen. Verder kan bij oudere

Les données les plus intéressantes pour nous sont enregistrées dans les chaînes GPRMC et GPGGA. Date, temps, position et hauteur sont des données disponibles. Par ailleurs, il est intéressant de connaître la fiabilité de ces informations. Les informations concernant la validité du message sont les suivantes (A ou V) : les données de la chaîne GPGGA (FIX = 0 : information insuffisante sur la validité du calcul, FIX >1 : calcul fiable) ainsi que le HDOP (Horizontal Dilution Of Precision). En plus, le nombre de satellites, dont les données sont reçues par le récepteur, constitue aussi une bonne information. Notez bien aussi que les données de position sont fournies en degrés et minutes décimales. Il semble aussi que le nombre de chiffres après la virgule ne soit pas le même pour tous les récepteurs GPS.

L'écran GPS de l'Arduino

Il est intéressant et instructif de connecter le récepteur GPS à un Arduino Uno ou Nano et d'observer les chaînes de données sur l'écran d'un PC.
Il existe beaucoup de récepteurs GPS pouvant être raccordés à un Arduino. Les circuits imprimés avec l'indication GY-NEO6MV2 sont très populaires et bon marchés sur internet. Ces circuits imprimés avec un CI u-blox, sont livrés avec une antenne active. Les messages NMEA sont disponibles, avec une vitesse standard de 9600 bits/sec, à la sortie TX du circuit imprimé.

On peut aussi utiliser l'ancien récepteur GPS Rockwell Jupiter. Ces récepteurs sont (étaient) très appréciés par les radioamateurs car ils étaient disponibles et bon marchés ; et aussi parce qu'ils délivraient un signal de sortie à 10 kHz pouvant être utilisé pour une synchronisation. Les connexions du Rockwell Jupiter sont bien décrites sur http://www.gpskit.nl/. La vitesse standard à la sortie NMEA est ici de 4800 bits/sec.



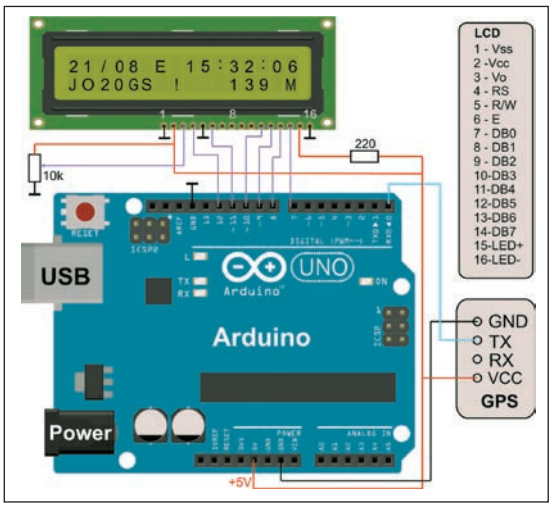
Afb. 2. Arduino met GPS-ontvanger

Fig. 2. Arduino avec récepteur GPS

Démarrez le programme Arduino IDE sur le PC et chargez le programme 'GPS-string' vers le moniteur série. Ce programme est téléchargeable à partir de : http://on4cdu.net/arduino-en-gps-data/ et ensuite : Arduino GPS software. Introduisez la vitesse des données pour le récepteur GPS (Jupiter 4800 et u-blox 9600 bits/sec) et chargez le programme dans l'Arduino. Raccordez maintenant la sortie NMEA du récepteur à l'entrée de l'UART de l'Arduino (pin RX pour Uno ; RX0 pour Nano). Les données GPS doivent maintenant s'afficher sur l'écran du moniteur série. N'oubliez pas de régler le moniteur série sur la bonne vitesse pour les données. Lorsqu'un programme doit être chargé dans l'Arduino, il faut débrancher le récepteur de la broche RX de l'Arduino, cette entrée servant aussi à charger un nouveau programme !
Si tout est en ordre, les chaînes GPS sont visibles sur l'écran. Il est intéressant d'étudier et d'interpréter les données. Débranchez l'antenne GPS et regardez alors ce qui se passe.
Les récepteurs modernes démarrent beaucoup plus rapidement et réagissent aussi plus rapidement à des variations de conditions que les anciens

Jupiter-ontvangers een onjuiste datum op het scherm verschijnen. Daarover later meer.
We weten nu dat de GPS ontvanger NMEA data aan de Arduino levert, en de volgende stap kan worden gezet.
De Roverbox, ontworpen door Christophe ON4IY, is een GPS-informatie display unit. We hebben hem binnen ons microgolfroverstation jarenlang gebruikt.
De originele Roverbox staat beschreven op http://www.qslnet.de/member/on4iy/roverbox.html.
Van dat ontwerp heb ik nu een Arduino-versie gemaakt. De nieuwe Roverbox bestaat uit een Arduino, een 16x2 LCD en een GPS ontvanger.

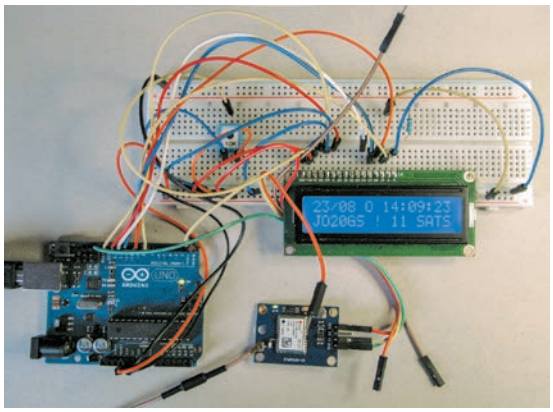
Voor de experimentele opstelling is geen externe voeding nodig; de USB aansluiting is de voedingsbron. Er mag natuurlijk wel een externe voeding gebruikt worden. De Uno is zo beveiligd dat er geen voedingconflict optreedt als er zowel een USB als een externe voeding is aangesloten.
De software genaamd 'GPS display' wordt in de IDE geladen, de interfacesnelheid wordt eventueel aangepast en de software wordt naar de Arduino overgebracht. Nu (en niet eerder) wordt de GPS ontvanger aangesloten. Als het goed is, zal nu data op het LCD te zien zijn. Op de eerste regel de datum, 'O' of 'E' (odd of even), en de tijd. Regel twee begint met de QTH-locator, dan een '?' als de data niet betrouwbaar is of een '!' als de data betrouwbaar is, en dan afwisselend de HDOP, de hoogte ASL, het aantal gebruikte satellieten en de positie van de zon.
De odd/even indicatie werd soms gebruikt in moeilijke microgolf QSO's, waarin afgesproken werd wie tijdens de even of oneven minuten zendt of ontvangt. Voor 'normale' microgolf vliegtuigscatter verbindingen is dit mechanisme echter te traag en dus niet bruikbaar. De waarde van de HDOP is een maat voor de kwaliteit van de positie-informatie.



Afb. 3. Schema van de experimentele GPS Display Unit

Fig. 3. Schéma de l'unité d'affichage GPS expérimentale

Bij toepassing van een Jupiter zal de tijd iets achterlopen. Dit is een gevolg van de procestijd in de ontvanger. Een correctie (in de software de variabele 'timeoffset') van 2 seconden is nodig. Verder kan het bij een Jupiter gebeuren dat er een onjuiste datum wordt weergegeven. Dit is een soort 'millenniumbug': de teller voor de datum heeft bij een aantal types een beperkte grootte, en is daardoor na 1024 weken (ruim 19 jaar) weer terug bij af. Om toch de juiste datum weer te geven moet eenmaal de juiste datum aan de ontvanger worden aangeboden, en deze informatie moet dan met een back-up batterij (3 V aan pin 3 van de twintigpolige connector) worden vastgehouden. De back-up batterij, bijvoorbeeld een CR2032 lithiumknoopcel, is bovendien nuttig omdat de ontvanger nu ook sneller start.
De informatie die aan de ontvanger moet worden toegevoerd staat ook in de software. In de set-up van het programma staat een regel met Serial.println. Maak deze regel actief en pas bij voorkeur ook de datum in deze regel aan. Laad het programma in de Uno en sluit dan op punt 0 (RX Arduino) en 1 (TX Arduino) van de Uno respectievelijk de NMEA-uitgang (pin 11 van de twintigpolige connector) en ingang (pin 12) van de Jupiter aan. Druk op de resetknop van de Uno om het programma opnieuw te starten, en de datum moet nu gecorrigeerd zijn.
Dit hoeft maar een keer te gebeuren als er een back-up batterij op de Jupiter is aangesloten. Maak daarna de regel weer onzichtbaar voor het programma of verbreek de verbinding met pin 12 van de Jupiter.



Experimentele opstelling van de GPS Display Unit met de NEO GPS-module

Montage expérimental de l'unité d'affichage GPS avec le module GPS NEO

Jupiter. Pour cette raison, ils sont aussi adaptés pour les applications de 'flight control'. De plus, avec les anciens récepteurs Jupiter, une date incorrecte peut apparaître sur l'écran. Nous reviendrons là-dessus plus loin.
Nous savons maintenant que le récepteur GPS fournit les données NMEA à l'Arduino, nous pouvons donc passer à l'étape suivante.
La Roverbox, conçue par Christophe ON4IY, est une unité d'affichage de l'information GPS. Nous l'avons utilisée pendant des années dans notre roverstation micro ondes.
La Roverbox originale est décrite ici : http://www.qslnet.de/member/on4iy/roverbox.html.
Maintenant, j'ai réalisé une version Arduino de ce projet. La nouvelle Roverbox est composée d'un Arduino, un afficheur LCD 16x2 et un récepteur GPS.

Pour la réalisation expérimentale, une alimentation externe n'est pas nécessaire ; celle-ci se fait via la connexion USB. Bien sûr, rien n'empêche d'utiliser une alimentation externe. L'Uno est protégé contre les conflits d'alimentation lorsque l'on utilise une alimentation externe ou par USB.
Le programme nommé 'GPS display' est chargé dans l'IDE, la vitesse de l'interface est éventuellement adaptée et le programme est transféré vers l'Arduino. Maintenant (et seulement maintenant) le récepteur GPS est raccordé. Si tout est en ordre, les données GPS apparaîtront sur l'afficheur LCD. Sur la première ligne, les données'O' ou 'E' (odd ou even) et le temps. La deuxième ligne commence avec le QTH-locator, suivi par un '?' si la donnée n'est pas fiable ou un '!' si la donnée est fiable. Ensuite, suivent le HDOP, la hauteur ASL, le nombre de satellites utilisés et la position du soleil.
L'indication odd/even a parfois été utilisée dans des QSO micro ondes difficiles, dans lesquels on précise qui émet ou reçoit pendant les minutes paires ou impaires. Pour les liaisons micro ondes par diffusion sur les avions, cette méthode est cependant inutilisable car trop lente. La valeur du HDOP est une mesure de la qualité de l'information sur la position.

Dans le cas d'un Jupiter, il y aura un peu de retard sur le temps. Ceci est une conséquence du traitement du temps dans le récepteur. Une correction (la variable 'timeoffset' dans le programme) de 2 s est nécessaire. Par ailleurs, il se peut que dans le cas de l'utilisation d'un Jupiter, la date fournie soit incorrecte. C'est une sorte de 'millenniumbug' : le compteur pour la date a, pour un certain nombre de types, une grandeur limitée et est remis à zéro après 1024 semaines (plus de 19 ans). Pour obtenir quand même la date exacte, il est nécessaire de donner une fois la date exacte au récepteur, cette information devant alors être conservée avec une pile de back-up (3 V à la broche 3 du connecteur à 20 broches). De plus, la pile de back-up, par exemple une pile bouton au lithium CR2032, est nécessaire car elle permet un démarrage plus rapide du récepteur.
L'information qui doit être envoyée au récepteur se trouve aussi dans le programme. Dans le set-up du programme, il y a une ligne avec Serial.println. Activez cette ligne et de préférence, ajustez aussi la date dans cette ligne. Chargez le programme dans l'Uno et raccordez au point 0 (RX Arduino) et 1 (TX Arduino) de l'Uno, respectivement la sortie NMEA (pin 11 du connecteur à 20 broches) et l'entrée (pin 12) du Jupiter. Appuyez sur le bouton de reset de l'Uno pour redémarrer le programme, la date doit maintenant être corrigée.
Cette procédure ne doit être appliquée qu'une fois si une pile de back-up est connectée au Jupiter. Rendez ensuite la ligne invisible pour le programme ou supprimez la liaison avec la broche 12 du Jupiter.

Denk erom de ontvanger telkens los te koppelen bij het laden van het programma via de USB aansluiting, want deze aansluiting maakt ook gebruik van de punten 0 en 1 van de Uno. Wie wat meer ervaring heeft kan ook het programma 'datum correctie Jupiter' gebruiken. Veel informatie over het configureren van de Jupiter ontvanger is te vinden op http://www.jrmiller.demon.co.uk/projects/ministd/jupcom.zip.

Enkele woorden over de software. Het maken van een programma om wat GPS-data op een lcd te schrijven is relatief eenvoudig. De moeilijkheden kwamen toen de antenne afgekoppeld werd en er onzin op het schermpje verscheen. Logisch, want ik had een vast formaat voor de datavelden van de strings aangenomen. Verder bleek ook dat de formaten van de berichten van een NEO en een Jupiter verschillen. In de software is hiermee nu rekening gehouden door telkens netjes de informatie tussen de scheidingskomma's te lezen, en niet van een vast aantal karakters van de data uit te gaan.
In de display unit worden alleen de GPRMC en GPGGA strings gebruikt. Iedereen is natuurlijk vrij om ook andere datastrings uit te lezen.
De software is opgesplitst in subroutines. De subroutine getField distilleert een bepaald veld uit een string, de subroutine locator berekent uit de positie de QTH-locator, en de belangrijke subroutine displayGPS bepaalt wat er op het lcd te zien is.
De berekening van de locator wijst zichzelf uit. Tijdens mijn vakantie is de routine op een aantal plaatsen in Europa getest en hij gaf de juiste QTH-locator weer. Een test aan de andere kant van de nulmeridiaan of op het zuidelijk halfrond is niet gebeurd. De routine om de positie van de zon te berekenen is 'sunpos'.

Wordt vervolgd

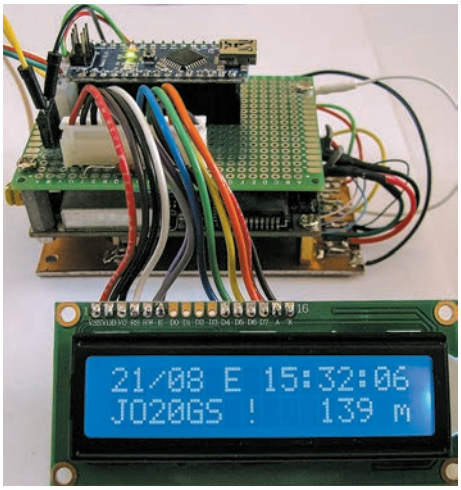
Hans Wagemans ON4CDU on4cdu@uba.be

Pensez à débrancher le récepteur chaque fois que vous chargez un programme via la connexion USB, celle-ci utilise aussi les points 0 et 1 de l'Uno. Les OM plus expérimentés pourrons aussi utiliser le programme 'correction des données Jupiter'. On trouvera beaucoup d'informations sur la configuration du récepteur Jupiter sur http://www.jrmiller.demon.co.uk/projects/ministd/jupcom.zip.

Quelques mots sur le software. La réalisation d'un programme pour écrire quelques données GPS sur un écran LCD, est relativement simple. Les difficultés sont survenues lorsque l'antenne fut branchée et que des bizarreries apparurent sur l'écran. C'était logique, car j'avais pris un format fixe pour les champs de données des chaînes. Par ailleurs, il est apparu que les formats des messages d'un NEO diffèrent de ceux d'un Jupiter. Dans le programme actuel, on tient compte de cela en lisant chaque fois convenablement l'information entre les virgules de séparation, et pas en considérant simplement un nombre fixe de caractères des données.
Dans l'unité d'affichage, on utilise uniquement les chaînes GPRMC et GPGGA. On peut bien sûr si on le désire, lire d'autres chaînes de données.
Le programme est scindé en sous routines. La sous routine getField extrait un champ déterminé d'une chaîne, la sous routine locator calcule le QTH-locator à partir de la position, et la sous routine importante displayGPS détermine ce qui doit être vu sur l'afficheur LCD.
Le calcul du locator est remarquable. Pendant mes vacances, la routine a été testée à un certain nombre de places en Europe et, à chaque fois, le QTH-locator exact a été obtenu. Elle n'a pas été testée de l'autre côté du méridien de Greenwich ni dans l'hémisphère sud. La routine 'sunpos' permet de calculer la position du soleil.

A suivre

Hans Wagemans ON4CDU on4cdu@uba.be



Een sandwich bestaande uit een printje met de Arduino Nano, een Jupiter GPS-ontvanger en onderop een printplaatje met een back-upbatterij en een 5V-stabilisator voor de Jupiter. Op de voorgrond een 16x2 lcd.

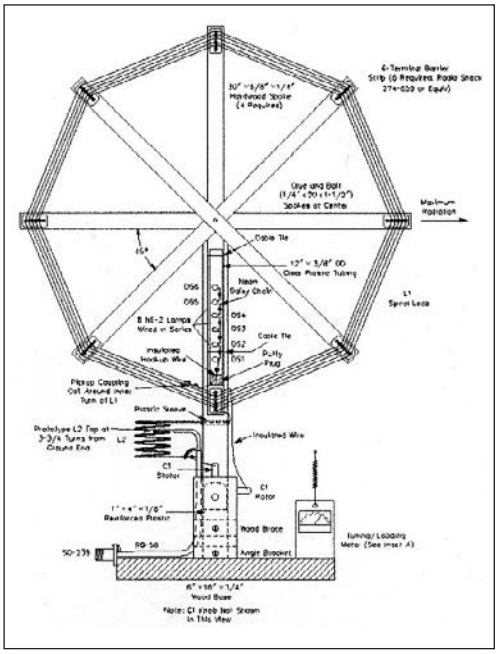
Un sandwich constitué d'un circuit imprimé avec l'Arduino Nano, un récepteur GPS Jupiter et en bas, un circuit imprimé avec une pile de back-up et un régulateur 5 V pour le Jupiter. A l'avant plan, un afficheur lcd de 16x2.

QRV op 80 meterband met een tafelantenne

door ON4DHL

Beste radiovriendinnen en vrienden, sommigen onder u weten dat ik me bezig hou met compacte antennen, omdat ik sowieso geen antennen mag plaatsen. Maar van een nood maakt men een deugd.

Enkele jaren geleden kreeg ik van collega's van sectie OSA een aantal schema's en fotokopieën van magnetisch veld loopantennen, waaronder die van G2ZBQ.
Deze man bleek in het verhaal veel onderweg te zijn en had een kleine loop gefabriceerd om mee op hotel te nemen. Deze bestaat uit een houten kader voor de loop en een houten voetstuk om de condensator te monteren.
Bon, ik heb deze toen in een aantal versies nagebouwd en hier geef ik wat toelichting.



G2BZQ 80 Meterband loop schema

Schéma de la loop 80m de G2BZQ

QRV sur la bande des 80m avec une antenne de table

par ON4DHL – traduit par ON5FM

Chers amies et amis de la radio, certains d'entre vous savent que je m'occupe d'antennes compactes suite au fait que je ne peux pas placer d'antenne normales chez moi. Mais d'une nécessité, on fait une vertu.

Il y a quelques années, j'ai reçu d'un ami de la section OSA, une série de schémas et de photocopies d'une antenne-cadre magnétique qui avait été développée par G2ZBQ.
Cet OM semblait un connaisseur dans ce domaine et il avait fabriqué une petite antenne cadre pour emporter avec lui lorsqu'il séjournait à l'hôtel.
Celle-ci se compose d'un cadre en bois pour la boucle et d'un pied, en bois également, où est monté le condensateur.
J'ai construit cette antenne en plusieurs versions et voici quelques explications.

Après la version avec un bobinage, je l'ai adaptée avec deux condensateurs variables ; un pour l'accord et l'autre pour l'adaptation de l'impédance à l'émetteur.