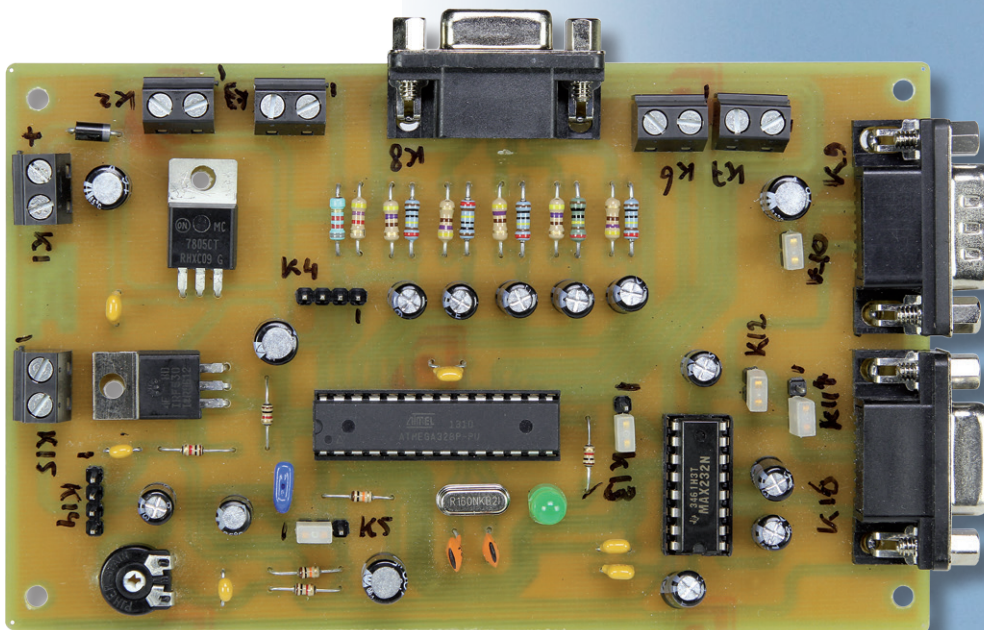


De zevende hemel... En hoger!

Universele data-acquisitiekaart voor een weerballon

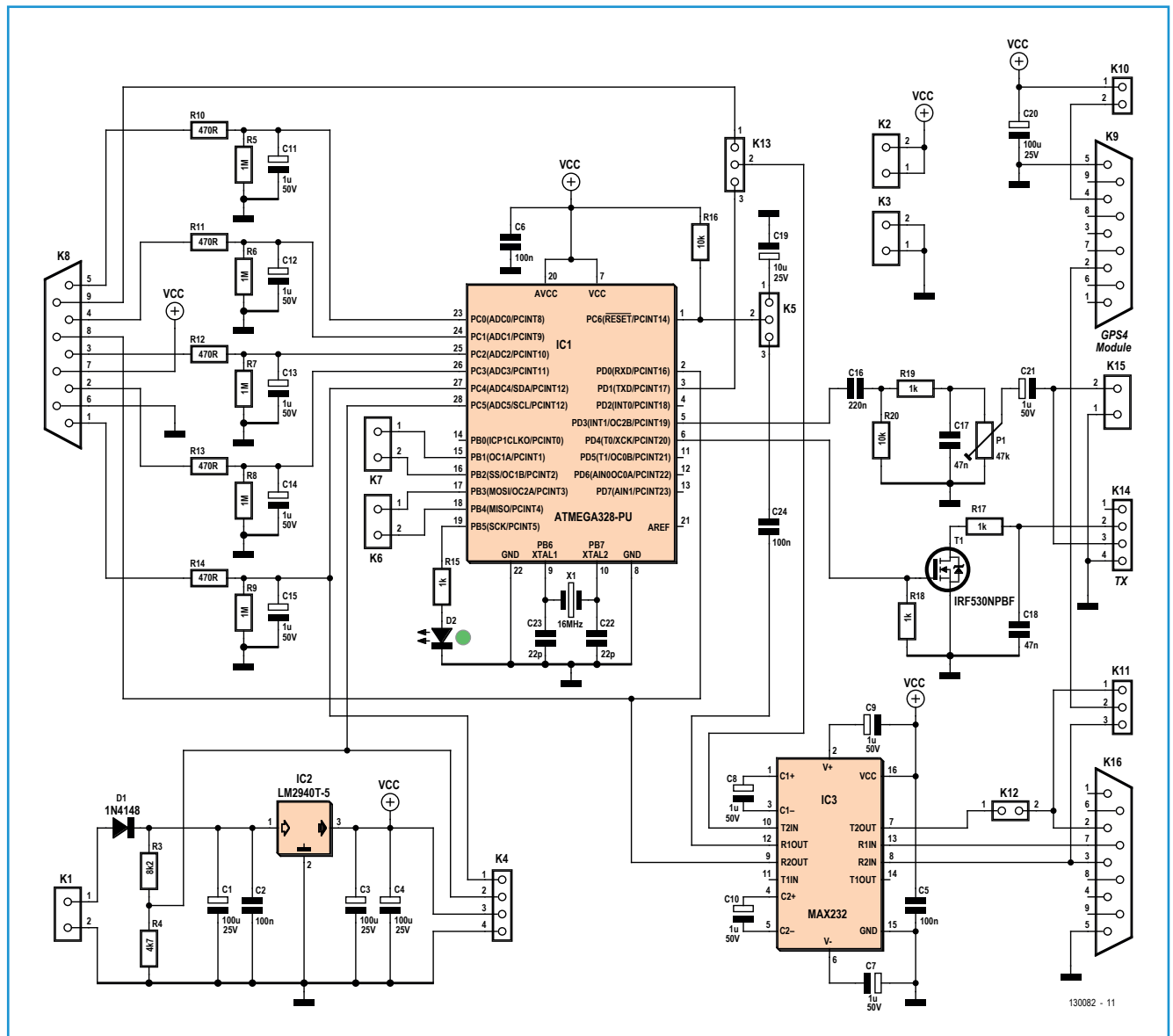
Dit verhaal gaat over regenwolken. Het verzenden van data vanuit een capsule die onder een weerballon bungelt is niet zo eenvoudig. Vaak is daar kostbare, geavanceerde apparatuur voor nodig. Gelukkig bestaan er goedkopere alternatieven. De kaart die we in dit artikel presenteren kan samen met een standaard radiozender en een GPS-ontvanger worden gebruikt om maximaal zes analoge signalen en positedata van een ballon naar de grond te sturen. En dat alles tegen heel lage kosten. We doen hier niet aan lucht-fietserij: Dit project heeft al twee succesvolle vluchten achter de rug!

Anthony Le Cren
(Frankrijk)



Eigenschappen

- Software op basis van Arduino
- Vijf analoge inputs op een 9-pens sub-D-connector (de zesde input meet de batterijspanning)
- Een 9-pens sub-D-connector voor GPS (NMEA0183A bij 4800 baud)
- RS232-poort voor in-circuit programmeren van de microcontroller (met Arduino bootloader)
- AFSK- en PTT-output naar de zender
- Maakt gebruik van APRS/AX25-protocol (packet radio)
- Vier inputs/outputs beschikbaar voor de gebruiker
- 5 V gestabiliseerde voedingsspanning
- Compatibel met Byonics-GPS4
- Compatibel met Baofeng UV-3R of UV-5R VHF-zend/ontvanger



Wat is er aan boord?

Het systeem in **figuur 1** is opgebouwd rondom een ATmega328P-microcontroller (MCU) met een klokfrequentie van 16 MHz. De software is gebaseerd op het open-source Arduino-project Trackuino [1]. Er kunnen veel sensoren op deze kaart worden aangesloten, maar er is toch nog ruim voldoende plaats voor uitbreidingen.

We zullen eerst de verschillende delen van het ontwerp bespreken aan de hand van het schema in **figuur 1** en het prototype van de auteur in **figuur 2** en **3**.

Analoge inputs

Er zijn vijf analoge inputs bedoeld voor het aansluiten van sensoren (temperatuur, druk, enz.). Deze zijn beschikbaar op K8, een 9-polige sub-D-connector (**figuur 2**, bovenaan). Na een laagdoorlaatfilter worden de signalen omgezet in digitale waarden door de A/D-converter (ADC) in de MCU, die beschikbaar is op poort C. De zesde ADC-input wordt gebruikt voor het meten van de batterijspanning. Spanningsdeeler R3/R4 reduceert de 9 volt batterijspanning naar een voor de MCU veilig niveau. Straks laten we wat voorbeelden zien van het uitlezen van een druksensor en een temperatuursensor met behulp van deze inputs.

Figuur 1. Schema van de data-acquisitiekaart voor een weerballon.

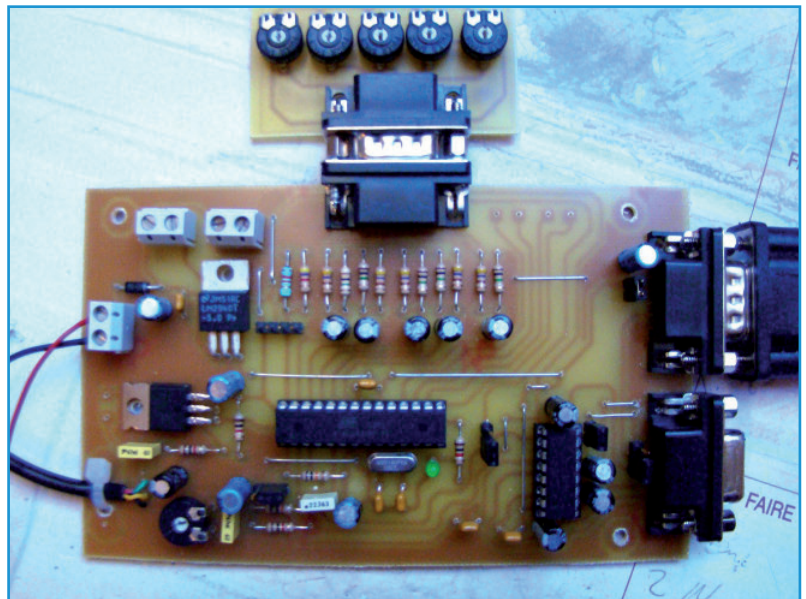
GPS

De GPS-eenheid wordt verbonden met K9 (figuur 2, de sub-D-connector rechtsboven). Zijn uitgangssignaal is beschikbaar op pen 2 van K11. Het is een RS-232-compatibel signaal, daarom wordt het via IC3, de klassieke MAX232, doorgegeven naar de MCU. K11 is bedoeld om het GPS-signaal òf naar de microcontroller (positie 2-3), òf naar de TX-pen van K16 (positie 1-2) te leiden. Zo kunnen we de tekstuele uitvoer van de NMEA0183A bekijken via een gewone seriële aansluiting. Vergeet daarbij niet jumper K12 te verwijderen, anders verstoort de TX2-output van IC3 het GPS-signaal.

De GPS-eenheid heeft 5 V nodig om te werken; dat wordt verzorgd door pen 4 van K9. De voedingsspanning kan worden onderbroken door de jumper op K10 te verwijderen. Dat is nuttig als we een simulator op een PC (bijvoorbeeld SIM GPS) gebruiken in plaats van een echte GPS. Een GPS-simulator helpt bij het ontwikkelen van de software, we kunnen daarmee de werking van het systeem controleren, voordat we het gaan oplaten. Verbind de kaart met de PC met een standaard nulmodemkabel (gekruste verbinding).

VHF-radio

Laten we nu kijken naar de interface met de packet-radio. De auteur adviseert een Baofeng UV-5R (figuur 3, links) of een UV-3R VHF-radio te gebruiken. Beide hebben een dubbele jack-connector (**zie figuur 4**). Er worden maar drie lijnen gebruikt: PTT (push to talk) (pen 2), MIC+ (pen 3) en GND (pen 4). De radio maakt gebruik van Frequency Shift Keying (FSK)-modulatie. Het audiosignaal schakelt dus heen-en-weer tussen twee frequenties (1200 Hz en 2200 Hz) om de binaire waarden te versturen. Omdat de MCU geen D/A-converter (DAC) heeft, wordt in plaats daarvan een pulsbreedtegemoduleerd (PWM)-signaal gebruikt. Dit PWM-signaal wordt omgezet in een quasi-sinusvormig signaal door een banddoorlaatfilter dat bestaat uit een eerste orde hoog- en laagdoorlaatfilter in serie. Het uitgangsniveau kan met potentiometer P1 worden aangepast, voordat het naar de MIC-input van de VHF-zender gaat. De radio wordt door een gesimuleerde PTT-schakelaar in zend-mode gezet. De radio heeft hier een input voor, dus de MCU kan de 'schakelaar' indrukken. Het PTT-signaal op



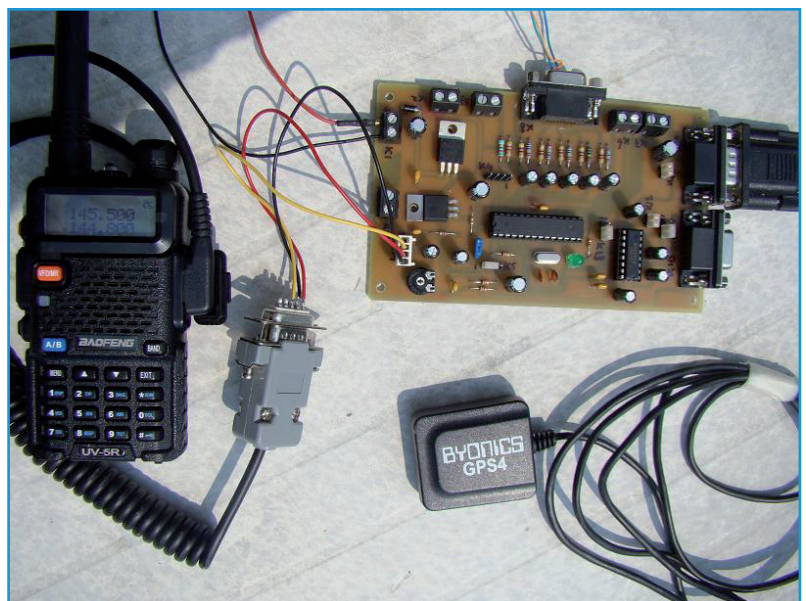
poortpen PD4 van de MCU stuurt een MOSFET van het type IRF530 aan, en vormt zo een open-drain-output die compatibel zou moeten zijn met de meeste VHF-radio's. Als PTT hoog is licht LED D2 op. Dat geeft aan dat er gegevens verzonden worden.

Figuur 2. De opgebouwde kaart van de auteur.

Voeding

De voeding is traditioneel opgezet. Het is niet aan te raden om een klassieke 7805-spanningsregelaar te gebruiken voor IC2, vanwege zijn enorme minimale spanningsval

Figuur 3. Complete opbouw van het systeem, inclusief de kaart, de VHF-radio en de GPS.

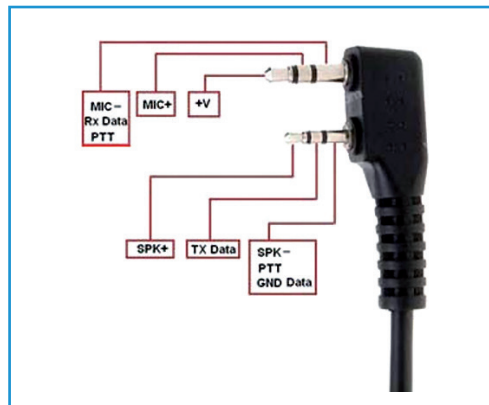


van ongeveer 3 V. Als we daar de spanning over diode D1 bij optellen, zou een batterijspanning van minstens 9 V nodig zijn. Als we kiezen voor een low-drop-regelaar zoals de LM2940T-5, kunnen we volstaan met een minimale voedingsspanning van 6 V. Het verschil van $9 - 6 = 3$ volt lijkt niet veel, maar de batterijspanning neemt duidelijk af bij de zeer lage temperatuur op grote hoogte. Om tijdens de hele vlucht voldoende spanning te garanderen, moeten we kiezen voor een voeding van minstens 8 tot 9 V. We kunnen die maken door zes batterijen van 1,5 V in serie te schakelen. Het totale stroomverbruik van de kaart is minder dan 60 mA, inclusief de GPS-eenheid.

Programmeren van de ATmega

De ATmega kan via connector K16 worden geprogrammeerd zonder hem van de kaart te halen. Sluit de kaart via een één-op-één seriële kabel aan op een PC met een echte RS232-poort. IC3, een klassieke MAX232, past de RS232-signaalniveaus aan voor de MCU. De TX- en RX-lijnen zijn beschikbaar op pen 2 en 3 van de MCU. Met K5 kan de bron van het resetsignaal voor de MCU worden gekozen. Zet een jumper op 2 & 3 om te resetten via de RTS-pen van K16. De PC kan dan met een reset de bootloader starten (dat regelt de Arduino-IDE voor ons).

De firmware is beschikbaar op de projectpagina bij Elektor.LABS [2], samen met de rest van de files. Zet na het programmeren de jumper op de pennen 1 & 2 om te voorkomen dat de MCU per ongeluk wordt gereset. Koppel tijdens het programmeren de GPS-eenheid, die dezelfde poort gebruikt, los van de MCU. Dat kan door hem los te koppelen van de kaart of door de jumper op K11 te verwijderen. **Tabel 1** geeft een overzicht van de jumperinstellingen voor elke situatie.



Figuur 4. De dubbele jack-connector van de Baofeng UV-5R VHF-radio. Er worden hier maar drie van de zes lijnen gebruikt.

Uitbreidingen

Als er nog meer elektronica aan boord van de capsule is, kan die via connectors K2 en K3 worden voorzien van 5 V voedingsspanning. Met behulp van de uitbreidingsconnectors K6 en K7 kan apparatuur tijdens de vlucht worden in- en uitgeschakeld. De firmware moet daarvoor wel worden aangepast! Connector K4 is toegevoegd voor communicatie via een I²C-bus. Hiervoor worden de poortpennen PC4 en PC5 van de microcontroller gebruikt. Dat gaat wel ten koste van twee van de analoge inputs (kanaal 4 en 5).

Tenslotte is er nog K13. Plaats hier een jumper op de pennen 1 & 2, om een seriële output op pen 9 van de analoge uitbreidingspoort K8 te maken. Op pen 8 van deze connector is een seriële input beschikbaar. Zo is het mogelijk om met een PC te communiceren via K8. Zet de jumper op pennen 2 & 3 van K13 om de seriële poort op K8 rechtstreeks met de MCU te verbinden. Deze extra poort kan handig zijn om de seriële lijn te koppelen met externe periferie, zoals een tweede microcontroller!

Opbouw

Het bestukken van de print zal geen problemen met zich meebrengen. Een aangepaste versie is in de vorm van Gerber-files, PDF's

Tabel 1: Jumper-instellingen

	MCU programmeren met de Arduino IDE	Start simulatie	Bekijken van GPS4-output	Lancering van de ballon
K5	2-3	1-2	1-2	1-2
K13	2-3	2-3	2-3	2-3
K12	Gesloten	Gesloten	Open	Gesloten
K11	Open	2-3	1-2	2-3
K10	Onbelangrijk	Open	Gesloten	Gesloten

zender. Dat is de rol van de Audio Frequency-Shift Keying (AFSK)-modulator. Deze zet de AX25-ASCII-string om in een serie toon-bursts met frequenties van 1200 Hz en 2200 Hz bij een snelheid van 1200 bits/s. Als de frequentie verandert tussen twee opeenvolgende bursts, wordt een logische 0 uitgezonden. En als de frequentie hetzelfde blijft, wordt een logische 1 uitgezonden. In **figuur 6** zien we een voorbeeld van hoe dat in zijn werk gaat. De software is compatibel met de huidige versie van Arduino (1.0.5). Vergeet niet het callsign van de ballon aan te passen in het bestand 'config.h':

```
#define S_CALLSIGN "F6XXX"
```

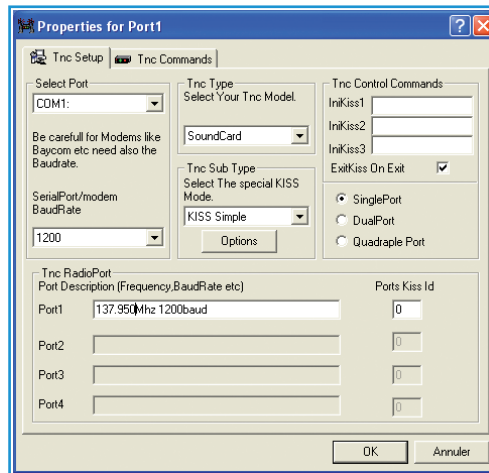
De gemakkelijkste manier om een geldig callsign te krijgen is contact op te nemen met een club van radio-amateurs in de buurt [5]. De code die het AX25-protocol implementeert is te vinden in het bestand 'aprs.cpp'. De functie `ax25_send_string` maakt de AX25-frames klaar voor verzenden en de functie `ax25_flush_frame` zorgt voor het verzenden zelf. Na het verzenden van de frames kan eigen code worden toegevoegd voor het besturen van extra apparaten die zijn aangesloten op K6 en K7.

Opzetten van de ontvangstketen

De AX25-frames van de ballon kunnen worden ontvangen met een afgestemde VHF/UHF-ontvanger of met een scanner. De audio-uitgang van de scanner wordt verbonden met de audio-ingang van een PC. De PC zorgt voor het decoderen.

De volgende software moeten worden geïnstalleerd:

- AGWPE: decoderen van de frames;
- AGWTrackerXP: weergave van de frames en de ruwe locatie van de ballon op een landkaart;



Figuur 7. Poortconfiguratie in AGWPE.

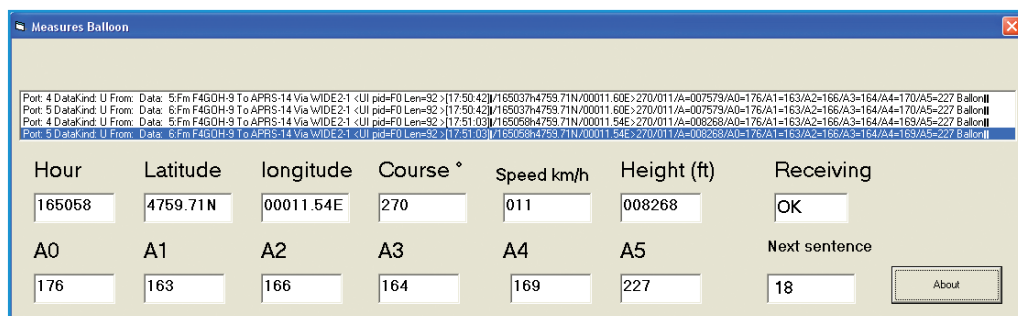
- Packet Engine: weergave van de gemeenten data.

Ook deze drie kunnen gratis gedownload worden op [6].

Start de PC opnieuw op na het installeren van AGWPE. Start het programma dan op en configureer een communicatiepoort, zoals weergegeven in figuur 7.

Installeer dan AGWTrackerXP. Voer het callsign in als ontvangen station. Er moet automatisch een TCP/IP-verbinding tussen AGWPE en AGWTrackerXP tot stand komen.

De laatste software die moet worden geïnstalleerd is Packet Engine. Dit tool slaat de ontvangen data op in een tekstbestand voor verdere analyse in een spreadsheet. Een screenshot is te zien in figuur 8. Laten we eens kijken naar één van die lange strings. Het is gemakkelijk om de AX25-header en het APRS-frame (met tijd, breedtegraad, lengtegraad enz.) te herkennen. De AX25-header eindigt met een verticale streep (|) en het APRS-frame eindigt na het woord "Ballon" en twee verticale strepen (||).



Figuur 8. Screenshot van Packet Engine met wat ontvangen data.

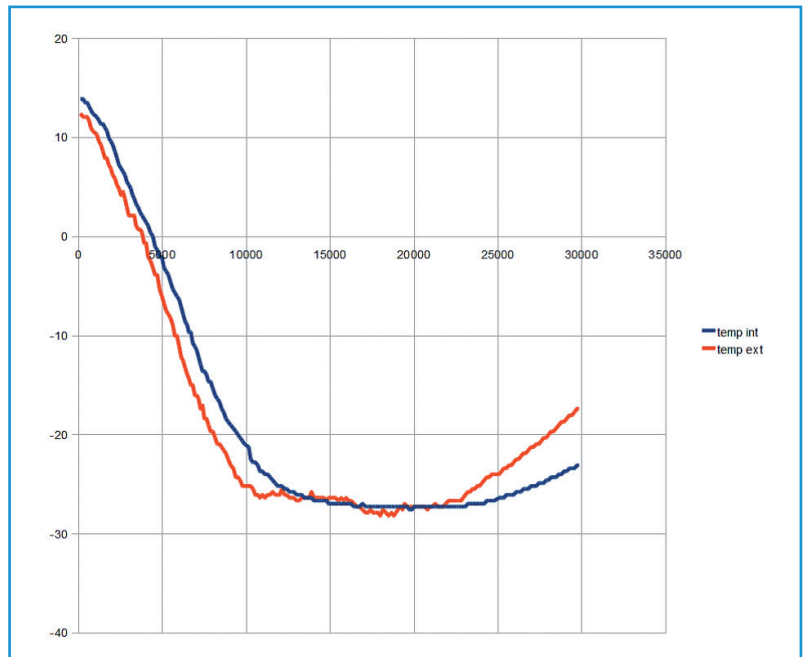
Ik wil je nooit meer kwijt

Tijdens de vlucht is de ballon vrij gemakkelijk te volgen. Maar als hij daalt tot beneden 2000 m, valt het radiosignaal meestal weg. Om zeker te weten dat de capsule terug te vinden is, is het verstandig een GSM/GPRS/GPS-tracker-module in te bouwen. Dat voegt niet veel gewicht toe en als hij is voorzien van een SIM kaart, kan hij SMS'jes met zijn lengte- en breedtegraad naar een mobiele telefoon sturen.

De antennes van GSM-masten zijn een beetje naar de grond gericht. Daardoor zal de tracker geen GSM-bereik hebben bij een hoogte van 700 m of meer. Gelukkig ontwaakt hij weer als hij een netwerk detecteert en dat is juist wanneer de capsule weer dicht bij de grond komt en landt.

Lanceren

De regelgeving over weerballonnen verschilt per land. Informeer u hierover goed voordat u met zo'n ballon aan de slag gaat, wat wel en niet is toegestaan en welke zendfrequenties mogen worden gebruikt. Heliumgas is behoorlijk duur. Daarom bouwen



veel hobbyisten een heteluchtballon of een zonneballon. In tegenstelling tot heliumbal- lonnen exploderen die niet; ze leggen vaak honderden kilometers af.

Figuur 9. Tijdens de testvlucht gere- gistreerde temperatuur. De meting moet zijn beïnvloed door de warmte van de cap- sule zelf, want de verwach- ting was een temperatuur van bijna 30 °C lager.

Checklist

Vorbereiding

- Start de software op de PC in deze volgorde:
 1. AGWPE
 2. AGWTrackerXP
 3. Packet Engine
- Schakel de VHF-ontvanger in (en stem af op de zender)
- Verbind de VHF-ontvanger nog niet met de PC

GPS-simulatie met SimGPS

- Controleer de jumpers op de kaart
- Verbind connector K9 met de PC via een getwiste kabel met een 9-pens female sub-D- connector aan beide kanten
- Start de NMEA-simulator, start het zenden van frames
- Schakel de VHF-zender in
- Voed de kaart met 8 à 9 V
- * LED D2 licht op als een frame wordt verzonden

Ontvangst

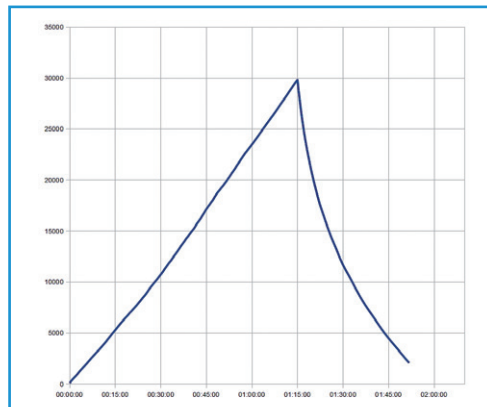
- Binnenkomende frames moeten te horen zijn op de ontvanger
- Als dat het geval is, sluit dan de ontvanger aan op de PC
- Nu moet het frame te zien zijn in AGWTrackerXP
- Klik met de rechter muisknop op het callsign en klik op 'Locate' en daarna op 'Show on map' om de positie van de ballon te zien op de landkaart.

Er zijn verschillende mogelijkheden voor het maken van foto's en video's. Vooral de kosten spelen hier een rol. De auteur heeft gewerkt met twee cameratypes: een HD-camera en simpele 'sleutelring-camera'. De eerste geeft afbeeldingen van heel goede kwaliteit, maar de tweede geeft ook leuke plaatjes tegen heel bescheiden kosten.

Een voorbeeld van de meetresultaten: de laagste buitentemperatuur die werd geregistreerd bij de experimenten van de auteur was ongeveer $-28\text{ }^{\circ}\text{C}$ bij een hoogte van 15.000 m, terwijl op 10.000 m een temperatuur van $-56\text{ }^{\circ}\text{C}$ werd verwacht. Blijkbaar werd de sensor beïnvloed door de warmte van de apparatuur in de capsule. Als de ballon de stratosfeer bereikt, neemt de temperatuur weer toe. De temperatuur binnenin de capsule bleek dezelfde tendens te volgen. In **figuur 9** zien we een grafische weergave van deze resultaten.

In **figuur 10** zien we de hoogte zoals gemeten door de GPS-eenheid. In het begin steeg de ballon met een vrij constante snelheid tussen 4 m/s en 6 m/s afhankelijk van het volume van het helium in de ballon. De ballon barstte op 30.000 m hoogte. Vandaar de zeer snelle afdaling.

Het hier beschreven systeem is ontwikkeld als een schoolproject. De auteur heeft veel plezier beleefd aan het, samen met zijn studenten, bouwen van het systeem. De druk op alle teamleden nam toe naarmate de lancingsdatum naderde, dus er werd zorgvuldig een checklist opgezet en die werd meerdere keren doorlopen, want er was maar één ballon. Als de ballon eenmaal is gestart, is het heel leuk om zijn reis live te volgen en de ontvangen data te decoderen. Na de vlucht komt de spanning over de positie waar de capsule is geland. Daarbij is de weersvoorspelling heel belangrijk. Als de ballon is geknapt, begint de jacht op de capsule. Hoewel de tracker het



Figuur 10. Door de GPS geregistreeerde hoogte. De ballon barstte op 30.000 m hoogte, dat is hier duidelijk te zien.

gemakkelijk maakt de capsule te lokaliseren, kan hij best hoog in een boom of zelfs in het water terecht gekomen zijn. Als de capsule gevonden is, is het weer leuk om te kijken naar de foto's en video's die tijdens de vlucht gemaakt zijn. Er is al een mooie verzameling plaatjes gepresenteerd in Elektor.POST #20, maar wie dat gemist heeft kan ze hier [7] alsnog vinden. Het laatste plaatje laat de hele route van de ballon na zijn vertrek uit Le Mans (Frankrijk) zien. Het is beslist de moeite waard om dit te bekijken!

(130082)

Weblinks

- [1] <http://www.trackuino.org>
- [2] <http://www.elektor-labs.com/130082>
- [3] http://en.wikipedia.org/wiki/Automatic_Packet_Reporting_System
- [4] http://www.tapr.org/pub_ax25.html
- [5] http://en.wikipedia.org/wiki/List_of_amateur_radio_organizations
- [6] <http://www.sv2agw.com/downloads/default.htm>
- [7] <http://bit.ly/17x8dIm>