

Dooie Fiets - Vervolg

Door Infant op zondag 7 april 2013 17:12 - Reacties (25)

Categorie: Gemod / Gefix, Views: 48.106

Voor de mensen die, net zoals ik, nog nooit een eclectische fiets gezien hebben, en wel eens benieuwd zijn hoe zo'n ding werkt, is deze ellenlange post wellicht interessant.

Dit is een vervolg op het intro van een paar dagen terug, en het volgende deel staat hier: Banana?



Overzicht van de elektronica in een elektrische fiets.

Als je op het plaatje klikt, zie je een fiets die ongeveer gelijk is aan wat ik gescoord heb.
(Mijne heeft geen versnellingen... bijvoorbeeld.)

In het eerste plaatje staan drie groene vlakken. Bij mijn fiets bevindt de motor zich achterin, de accu zit in het frame weg gewerkt, en op het stuur zit een klein display met twee knopjes en wat tellertjes. Er zijn ook fietsen met bijv. de motor voorin (geel) en de accu op het bagage rek. Er zijn zoveel fietsen, dat elke willekeurige combinatie denkbaar is.

Nou is deze fiets stuk, maar de vrij korte handleiding geeft duidelijk aan wat het zou moeten doen.

De gedachte achter deze fiets is dat als je begint met trappen, de motor mee 'trapt'. En dat maakt fietsen tegen de wind in, of een heuvel op, een stuk makkelijker.

Deze heeft die standen, Eco, Normaal en Power. En die hebben eigenlijk alleen betrekking op hoe hard de motor dus mee trapt. Hoe meer je zelf doet, hoe langer de accu mee gaat.

De fiets heeft ook twee trommel remmen, voor en achter. Met van die knijpertjes aan het stuur. En uiteraard twee lampen. Eigenlijk dus ook zonder elektronica, een prima fiets.

Waarom doet ie ut niet?

Als je wat rond leest, kom je er achter dat exact dezelfde elektronica in andere merken, o.a. Sparta, Koga Miyata, en Hercules, zit.

Als deze fietsen uiteindelijk stuk gaan, is de klachtomschrijving op fora meestal hetzelfde. Hij deed ineens niks meer.

Er zijn een paar mensen die na het kopen van een nieuwe accu, hun oude hebben open gemaakt. Bijna allemaal vertellen ze dat er corrosie op het printje in de accu zat. Meestal was er met de cellen zelf niks mis.

Dus, ik kocht deze fiets met de verwachting dat er een printje in de accu zou zitten die overleden is, en dat de cellen nog wel in orde waren.

Nou, openmaken die handel! Na wat gepeuter (wat vrij eenvoudig gaat), heb je een gigantische banaan-vormige accu in je handen, met wat stekkertjes er aan:



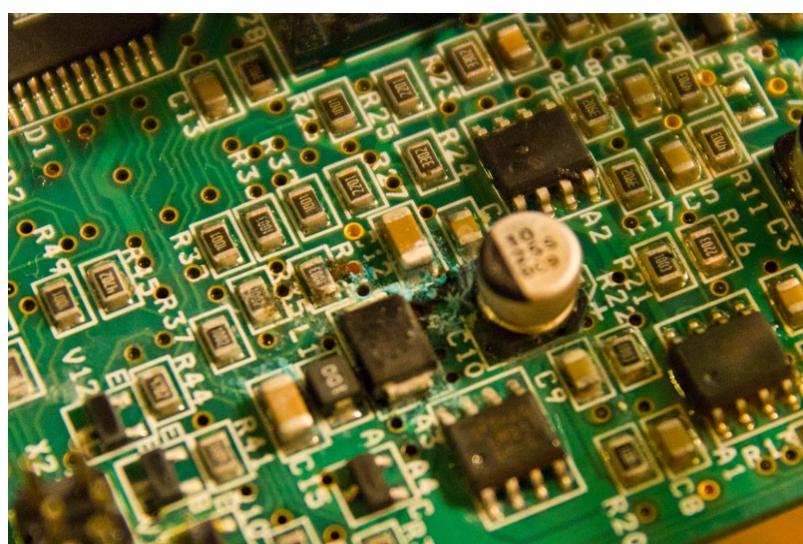
Een stock foto...

De hierop volgende chirurgische-operatie duurde wat langer, want de randjes zaten goed aan elkaar geplakt. Maar uiteindelijk wil het:



Accu na het open maken... en nog wat andere zooi.

En jawel, in deze accu zit exact hetzelfde PCBtje als ik verwachtte. En deze had duidelijk een klein beetje teveel vocht gezien:



Er zat helaas geen lader bij, maar dat geeft niet. Ik heb namelijk nog een stapeltje 1000VA UPSsen zonder accu's liggen. Normaal gaan daar 2 loodaccu's in, maar hier zijn ze ook zeer content mee:



De UPS vind alles best. Hij snapt toch niet dat het eigenlijk een fiets-accu is.

Helemaal aan het einde van deze post, staan foto's van de voor en achterkant van dit PCBtje, en een lijstje van de IC's die ik na wat moeitje heb kunnen aflezen.

Vanuit dit printje gaan 3 dikke dropveters naar de motor toe. En die zijn aardig logisch kleur gecodeerd, en gelabeld:

- BOW-PWR - Rood
- BOW-GND - Zwart
- DAT - Geel

Zwart hangt uiteraard direct aan de min van de accu. Geel gaat richting de motor, en tevens naar het schermpje op het stuur. Rood kan d.m.v. van een relais geschakeld worden.

Of dit relais aangaat, besluit waarschijnlijk het printje zelf. Er zit ook een stroom-meet

weerstand op. Dus dit printje zal kijken of de stroom en spanning die de accu levert, binnen acceptabele niveaus ligt. Als dat niet zo is, krijgt de motor geen prik.

Maar... Als de motor alleen aan of uit gezet kan worden.... hoe kunnen er dan 3 snelheid standen zijn?

Wat gesuggereerd wordt, is dat geel een 1-wire bus is, die naar de motor en naar het display gaat. Zo kan het display eigenlijk alleen als input fungeren om tegen de motor te zeggen hoe hard je wilt, en kan het accu printje tegen het display vertellen hoe vol de accu nog zit.

Dus, er zit duidelijk iets in de motor, wat de snelheid kan regelen.

De motor

Omdat dit geen cursus fietsen maken is, heb ik daar ook niet echt plaatjes van geschoten. Dus, vertrouw me als ik zeg, dat het achterwiel er vrij gemakkelijk af gaat.

De motor heeft aan weerszijde een aantal kruiskop schroeven zitten. Aan de ene kant verdwijnt het eerder beschreven 3-aderige draad langs de as de motor in, en aan de andere kant zit alleen een aluminium deksel met daarin dus een aantal schroeven.

Dat zag er eenvoudig uit om los te maken, dus daar begon ik. Je moet met redelijk wat kracht de deksel los peuteren. Hij klemt namelijk om een lager heen, maar uiteindelijk zie je dan dit:



Nu wordt het een beetje duidelijk wat er in zit. Het PCB aan deze kant is bevolkt met een MC68HC908MR32microcontroller, en er verdwijnen drie draadjes de motor in gelabeld HALL1 tm 3.

En het tweede google resultaat is een pdfje over een Control Board.

En de introductie daarvan leest:

Motorola's embedded motion control series MR32 motor control board
is
designed to provide control signals for 3-phase ac induction, 3-phase
brushless
dc (BLDC), and 3-phase switched reluctance (SR) motors.

Nou, dat lijkt me duidelijk. In het Nederlands is deze motor dus een 3-fase borstelloze DC motor.

Ik zou graag een linkje naar een wiki artikel plaatsen.... maar eigenlijk wordt je daar niet echt veel wijzer van.

Het is dus niet dit.

Deze vier regeltjes heb je ook niet zo veel aan.

Het is er zelfs zo sneu mee gesteld, dat ik moet concluderen dat het Duitse wiki artikel het beste is. (Er staat zelfs een plaatje van een fiets bij.)

Om meer duidelijkheid te creëren, gaan we verder met open slopen.

Omdat er flinke magneten in zitten, moet er aardig wat kracht uitgeoefend worden om het misbaksel ervan te overtuigen dat hij uit elkaar moet. Maar uiteindelijk....



Donut.

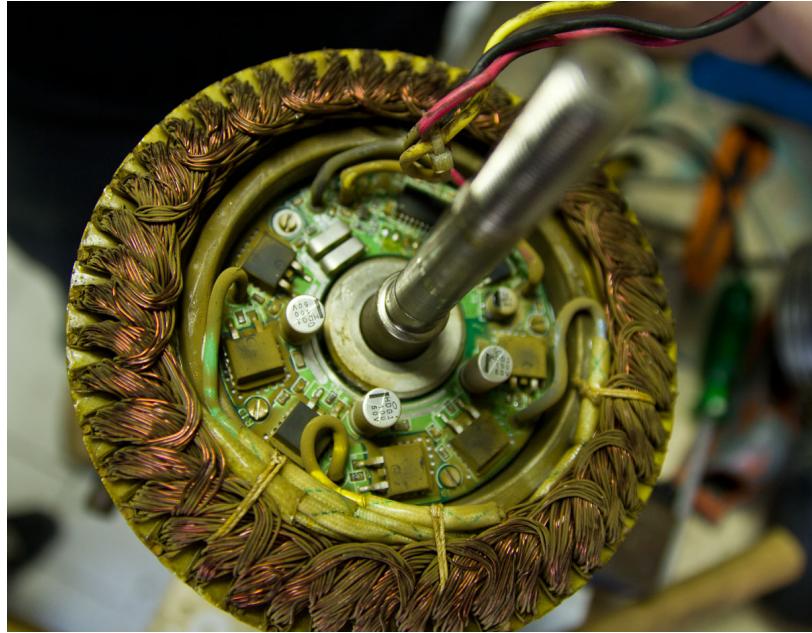
Wat je dan over hebt, is het wiel, met een ring van magneten aan de binnenkant.



Enigszins aangevreten magneten.

Het zit er naar uit dat die begonnen zijn zichzelf op te eten. De coating van de magneten is op sommige gaan los gelaten. Die (metalen)coating is gedurende gebruik tot stof vermaalt, en dat zit nu overal tussen.

De andere kant zit er dus ook uit alsof het net een week op een kurkdroog (metal)festival heeft mee gemaakt:



"Dat lijkt wel een desert, Storm."

Ik kan het nu toch niet laten om te zeggen: What the fuck.

Geen 1 van de drie PCB's die ik ben tegen gekomen had enige vorm van water afstotende coating. Hoe moeilijk was dat nou geweest? De motor heeft ook geen enkele vorm van vocht wering. Een o-ringnetje had voor die 2000 ballen toch wel gekund?

En verder is het idee van een borstel loze motor, dat er geen borstels zijn om koolstof-stof door je motor heen te blazen. Maar dat is nu vervangen door (waarschijnlijk) nikkel-coating-stof?

Ik hoop dat bij aanschaf van zo'n fiets, aanbevolen wordt niet in de regen te gaan fietsen.

Anyhow. Op het laatste PCBtje komt de aap uit de mouw, want hier zit het volgende IC'tje op: een [Intersil HIP4086AB](#).

En dat is, volgens de datasheet, een 80V, 500mA, 3-Phase MOSFET Driver. En de typical application ziet er zo uit:

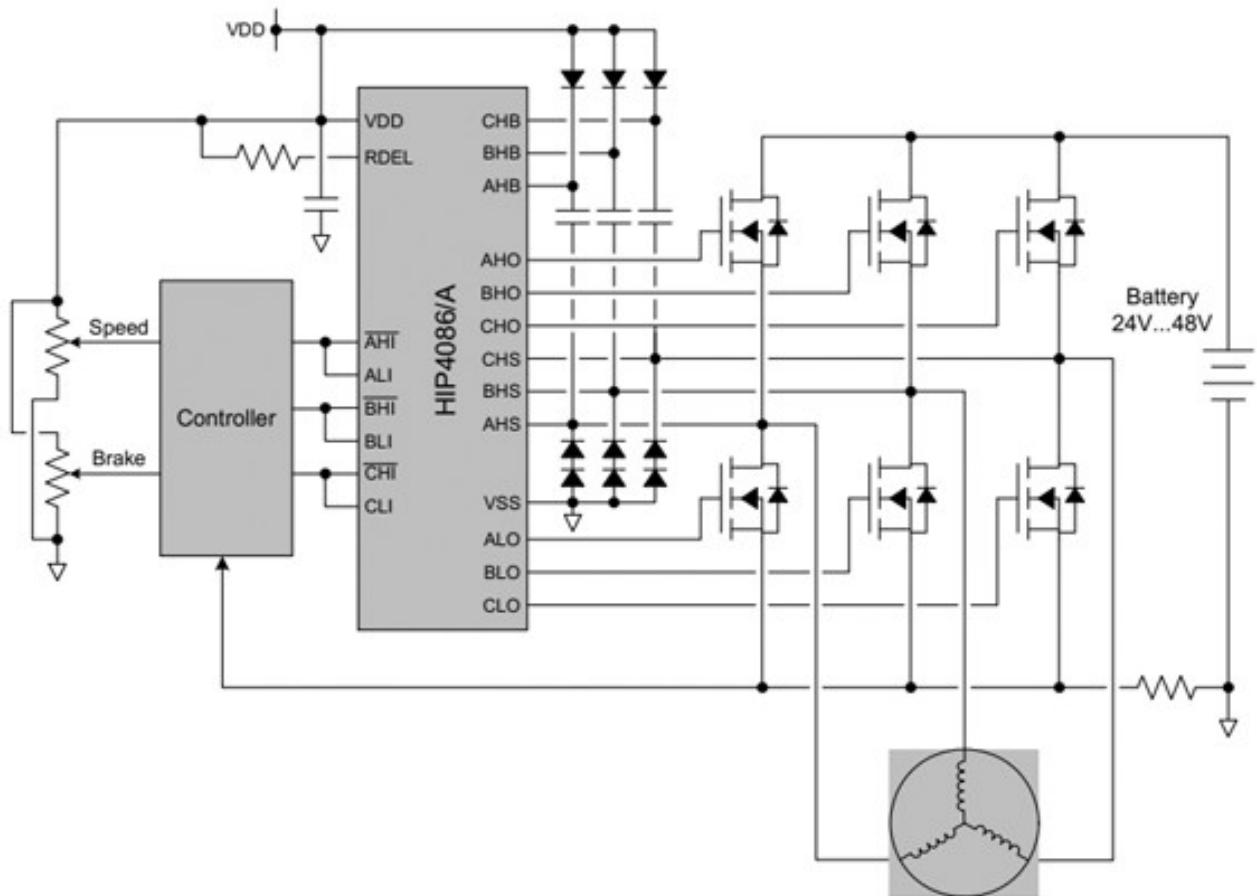
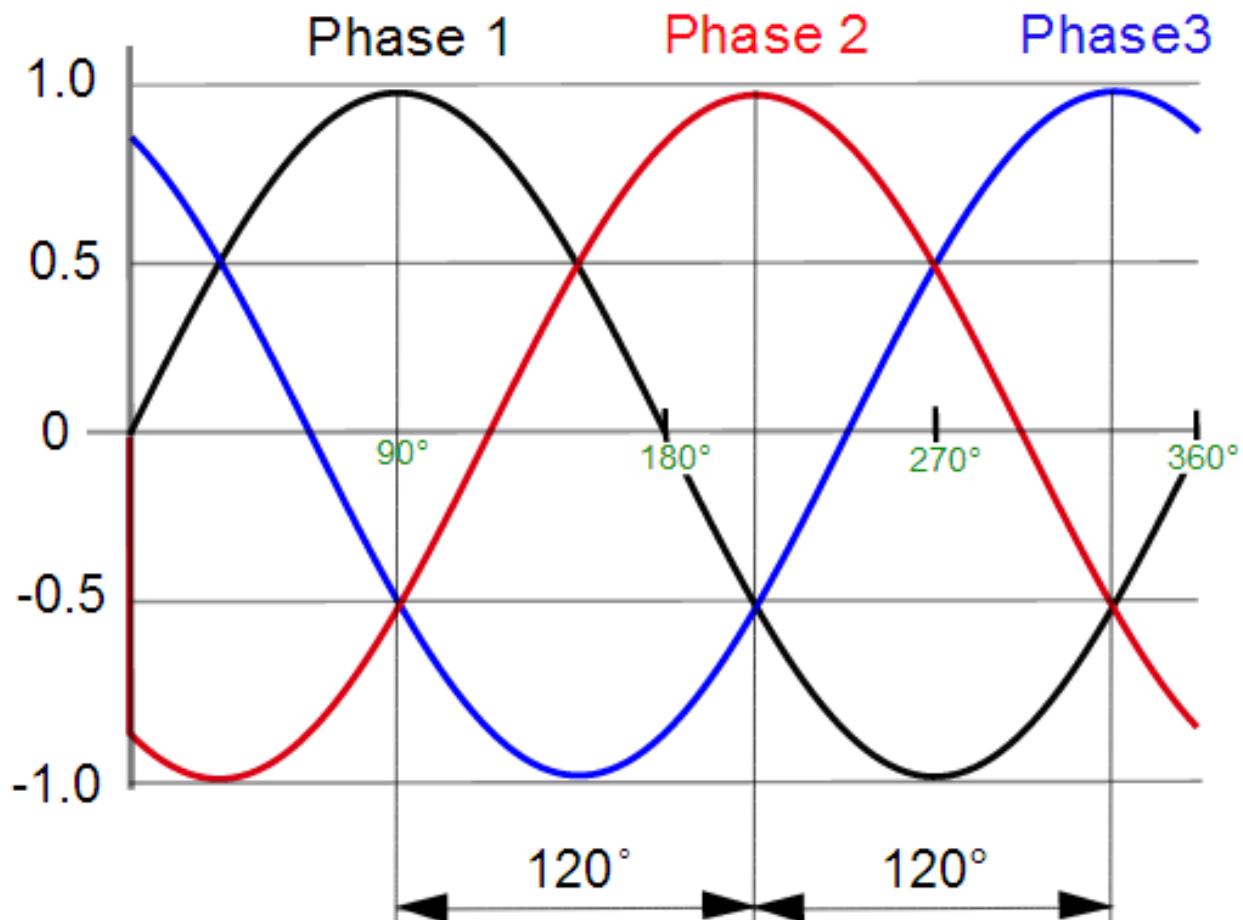


FIGURE 1. TYPICAL APPLICATION

En dat is dus ook hoe die hier toegepast is. De 6 mosfets zijn [IRF3205](#). Voor de rest zit er aan de onderkant een berg diodes, en nog wat los grut. Een 12V regelaar om VDD te maken. En dat was het.

Al het koperdraad wat je ziet, zit aangesloten in een zogenaamde Y, of ster configuratie. Door de mosfets op een handige manier aan te zetten, kan de stroom beide kanten kanten op door de drie windingen lopen.

Als je dus wilt dat de motor gaat draaien, duwt één winding de magneet van zich af, terwijl de tweede hem aantrekt. De derde duwt hem weer af. Dit proces wisselt zich steeds af, en de stroom door de spoelen zou er dan met een beetje fantasie ongeveer zo uit moeten gaan zien:



Als ik zou naar alle componenten kijk, is er niet echt een reden dat dit ding persé 24V nodig heeft. De driver vind alles tot 80V prima. De mosfets gaan stuk boven de 55V, en de condensator vind alles boven 50V niet leuk meer.

Eigenlijk kun je dus, net als bij laptops, een redelijk uiteenlopende spanning aanbieden. Ik zou hier zeggen dat alles tussen ongeveer 15V en 40V prima gaat werken.

Stel dus dat de accu echt versleten zou zijn. Een goedkope (maar niet passende) oplossing zou dan bijv. 3 stuks 7Ah loodaccutjes (c.a. 50 Euro inc. verzending) kunnen zijn. Je hebt dan ongeveer dezelfde capaciteit, c.a. 250Wh. En dus weer dezelfde actieradius. (De accu die erin zit is 24V, 10Ah. Dus 240Wh).

Wat kan het in theorie?

Nu het duidelijk is wat er in huis gehaald is, kun je gaan fantaseren wat het allemaal kan. De motor is vrij efficiënt, kan kortstondig een stuk meer vermogen dan 250W leveren. (Denk aan 500-1000W).

De motor kan vooruit/achteruit, hij kan zeer efficient als dynamo dienen. Je zou dus als je wilt volledig elektrisch kunnen remmen, en een vrij groot gedeelte van die energie kan hoppa, terug de accu in. Dat zou betekenen dat je nooit de mechanische remmen hoeft te vervangen.

Je kan hem als hometrainer gebruiken om 200-300W het stopcontact terug in te persen. (daar heb je eigenlijk alleen een extra omvormer voor nodig)

Je zou een snelheid kunnen instellen, bijv. 16.7km/u (of een ander willekeurig getal, you name it.), en dat houd dit ding dan keurig netjes aan. Je kunt hem gebruiken om je notebook mee op te laden als je op fiets vakantie gaat. Heck, als je 3 70Wh notebook accu's hebt, kun je die gewoon om op te rijden.

Dit moet met deze elektronica al te doen zijn.

Ik vind de motor best leuk in elkaar zitten. Er zit een strain-gauge om de as heen. Hiermee kan bepaald worden hoeveel moeite jij aan het doen bent, en dus of er trap ondersteuning moet komen. D.m.v. de 3 hall sensoren kan de snelheid bepaald worden. (En die wordt vervolgens door het gele draadje heen, naar het display gesmeten.)

Deze hall sensoren zijn trouwens nodig om bij stilstand, of heel langzaam rijden, te kunnen zien waar de magneten zich bevinden ten opzichte van de windingen.

Verder zou je bijvoorbeeld een rem hendel kunnen hebben, die bij harder knijpen steeds harder elektrisch remt. Of je kunt instellen dat je helemaal niet ondersteund wilt worden, maar steeds 10, of 20 of 100W de accu in wilt pompen.

De verlichting kan met een knopje aan/uit, en werkt gewoon van de accu of motor. Dus je hoeft ook nooit met een losse dynamo te kloten.

Waarom doet het dat niet dan?

Twee redenen: Wij hebben in Nederland een wet. Omdat fietsen levens gevaarlijk is, mag je zonder rijbewijs, helm, verzekering en kenteken alleen op een elektrische fiets rijden als die:

- Maximaal 250W kan leveren.
- Alleen hulp bied. Dus als jij stopt met trappen, moet de fiets stoppen met helpen.

- Maximaal 25 km/h kan.

Dat laatste kan ik mee leven, dat eerste lijkt me ook prima, maar de tweede regel is een beetje jammer.

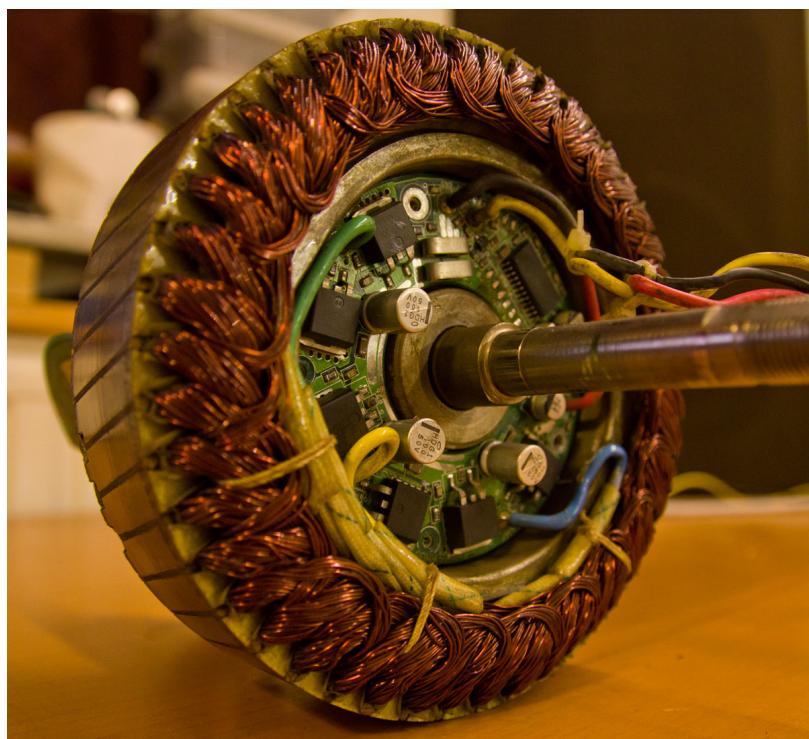
De rest die ik zomaar oplepelde, doet het niet omdat ze geen zin hadden om dat te maken.... denk ik.... Het achterlampje moet bijvoorbeeld een los batterijtje in.... WHY!!!!

Accel group, het overkoepelende holding waar Batavus en andere merken onder vallen, heeft kennelijk een hele berg triviale patenten liggen. De communicatie in deze fiets, tussen motor, accu en display... is dan ook goed dicht getimmerd. Niemand mag er aan zitten of iets aan veranderen. Dus elk wissewasje moet je voor naar de dealer toe, die het eigenlijk ook allemaal niet zo goed weet.

Er is kennelijk ook besloten om alle productie naar China te outsourcen. Altijd een goede beslissing als het gaat om betrouwbaarheid, milieu-vriendelijkheid, onderhoudsgemak en aanpasbaar-heid.

Conclusie:

Ik ga me niet vervelen deze zomer. Voor nu heb ik hem netjes schoon gemaakt...



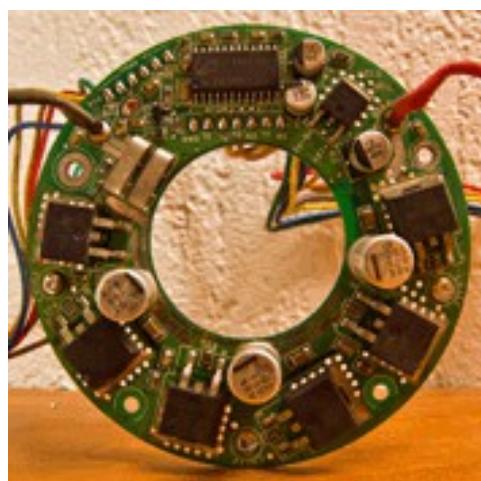
... en zonder electronica weer ingepakt. Alle crunchy geluidjes zijn nu weg, en je kunt lichtjes afremmen door 3 draadjes tegen het frame aan te drukken. (Dat is grappig.)

Ik moet zeggen dat hij een stuk lichter, en beter rijd dan mijn vorige fiets.

Ik ga nu verzinnen hoe ik dit kan combineren met het HP accu printje....waar ook nog binnenkort een post over zal verschijnen....

Het archief

Hieronder foto's van de ingewanden van een Batavus Padova Easy. De ronde PCB's komen uit de toprun motor, de hub motor die zich in het achterwiel bevind.



De drie PCBs. Klik voor grotere foto's van beide kanten.

De kleine IC's op het accu PCB zijn lastig te lezen, maar er staat op:

Atmega32L

MCP6071 (1 aan elke kant)

LM393N

LM293

En een 5V regelaar.

Banana?

Door Infant op maandag 9 juni 2014 12:37 - Reacties (16)

Categorieën: Gemod / Gefix, Sparta: Fiets verbouwen tot het stuk gaat., Views: 19.511

Ik acht mij niet verantwoordelijk voor hoge bloeddruk en/of ongecontroleerde woede aanvallen veroorzaakt door spellingfouten.

Dit is in vervolg op: [Dooie Fiets - Vervolg](#), en gaat hier verder: [Bananen Taal](#)

Eens in de zoveel tijd struin ik rond op Marktplaats, gewapend met mijn favoriete zoektermen. Afgelopen week vond ik dat eindelijk een defecte banaan, waar ik al een hele tijd naar op zoek was. (Uiteraard was dat niet de zoekterm.) Het betreft een accu in de vorm van een banaan, voor in mijn [kapotte zombie fiets](#).

Ik heb er drie gekocht, gezien ze zo schandalig goedkoop waren. Ze werden door een fietsen winkel aangeboden en ik kreeg er zelfs een mooie marge factuur bij opgestuurd. Ik blij, en de fietsenmaker in kwestie denk ik ook.

Hij neemt deze dingen in als iemand bij hem komt om nieuwe accu's te plaatsen.

Uiteraard mag zo'n klant dan 6 Euro verwijderingsbijdrage betalen bovenop de andere willekeurige kosten die hij in rekening gaat brengen. Normaliter moet hij dat bedrag weer afstaan aan de plek waar hij die dingen heen brengt, denk ik. De gemeente? Een donker plekje ergens in de bosjes? Chemisch afval?

In plaats daarvan krijgt hij nu nog meer geld, om het naar mij op te sturen. Een win situatie voor de fietsenmaker, een verlies-verlies-situatie voor de persoon die hem bij hem heeft ingeleverd.

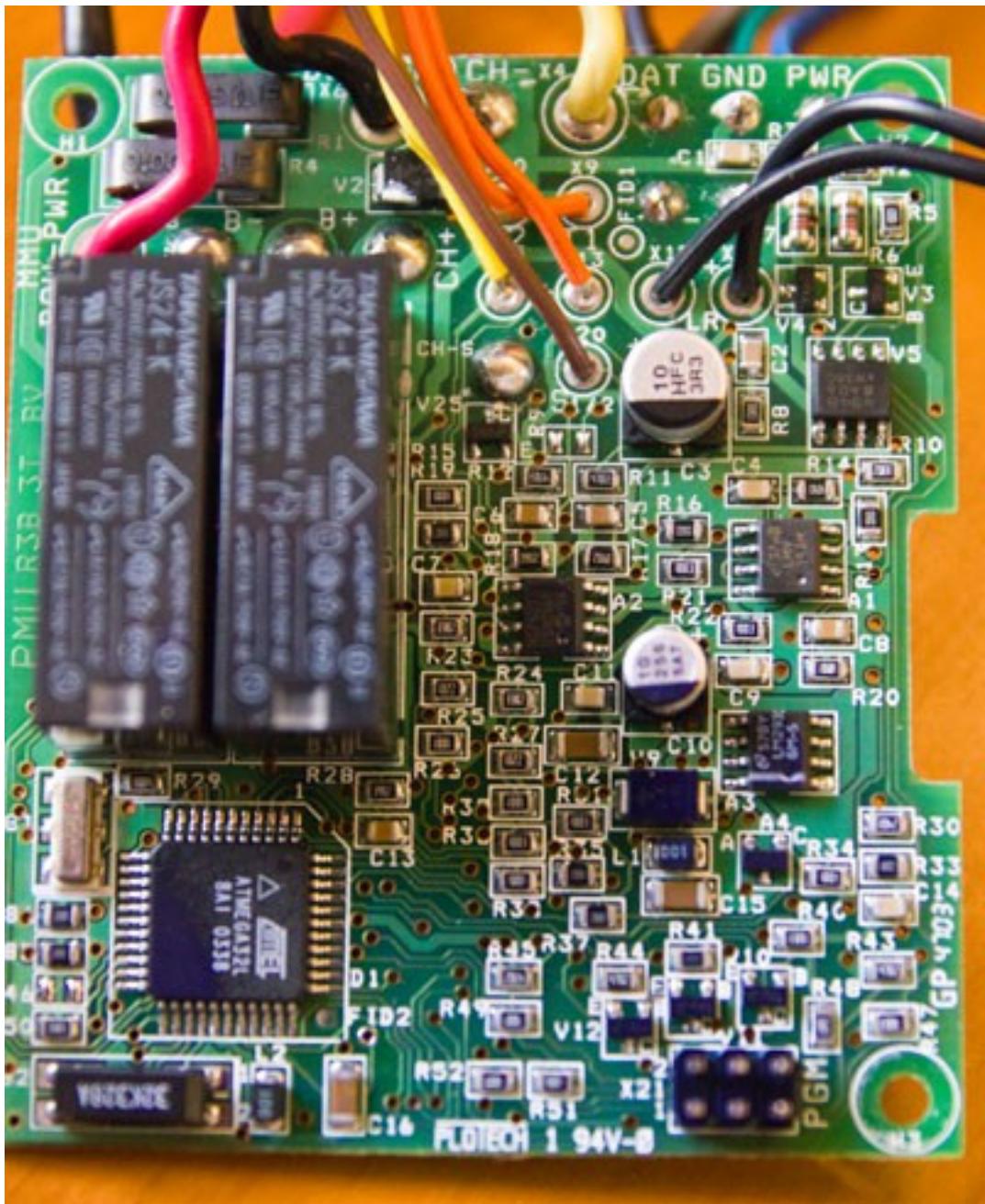
Na een paar dagen komt de TNT/PostNL met een berg langwerpige dozen aan zetten:



De eerste 30 seconden van dit filmpje beschrijven extreem exact hoe je je dan voelt.

Open maken!

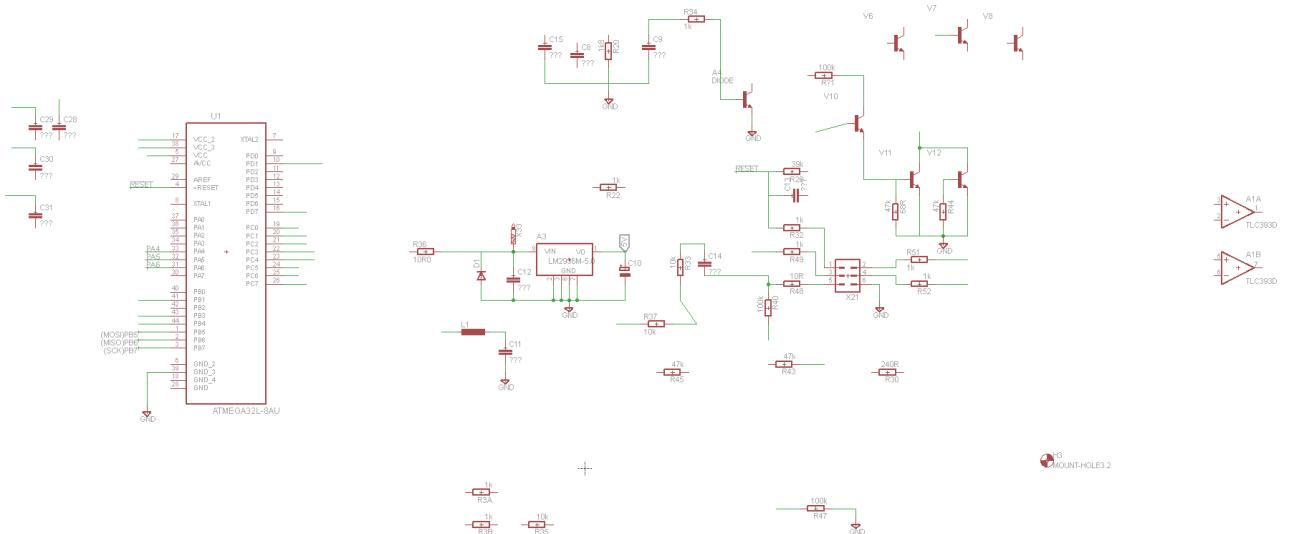
Na wat gepiel en gepeuter, gaat ook deze banaan open. Het gaat hier om een 9Ah accu, t.o.v. 10Ah , maar het printje is identiek aan degene die in mijn fiets zat. Het enige verschil is dat deze minder kapot lijkt, en helemaal geen coating heeft.



Uiteraard doet ook dit ding 10 keer niks.

Stiekem heb ik een tijdje geleden, toen ik me verveelde, een beginnetje gemaakt om het schema van dit printje te tekenen. Het is maar een twee-laags pcb, er zit alleen een idioot oerwoud aan analoge meuk op met weerstanden die overal overheen, naartoe en tussendoor gaan.

Ik dacht, ik begin bij de 6-pins connector, want die lijkt verdacht veel op de standaard 6-pins Atmel ISP header. Dit is de staat waarin ik het "schema" achter heb gelaten:



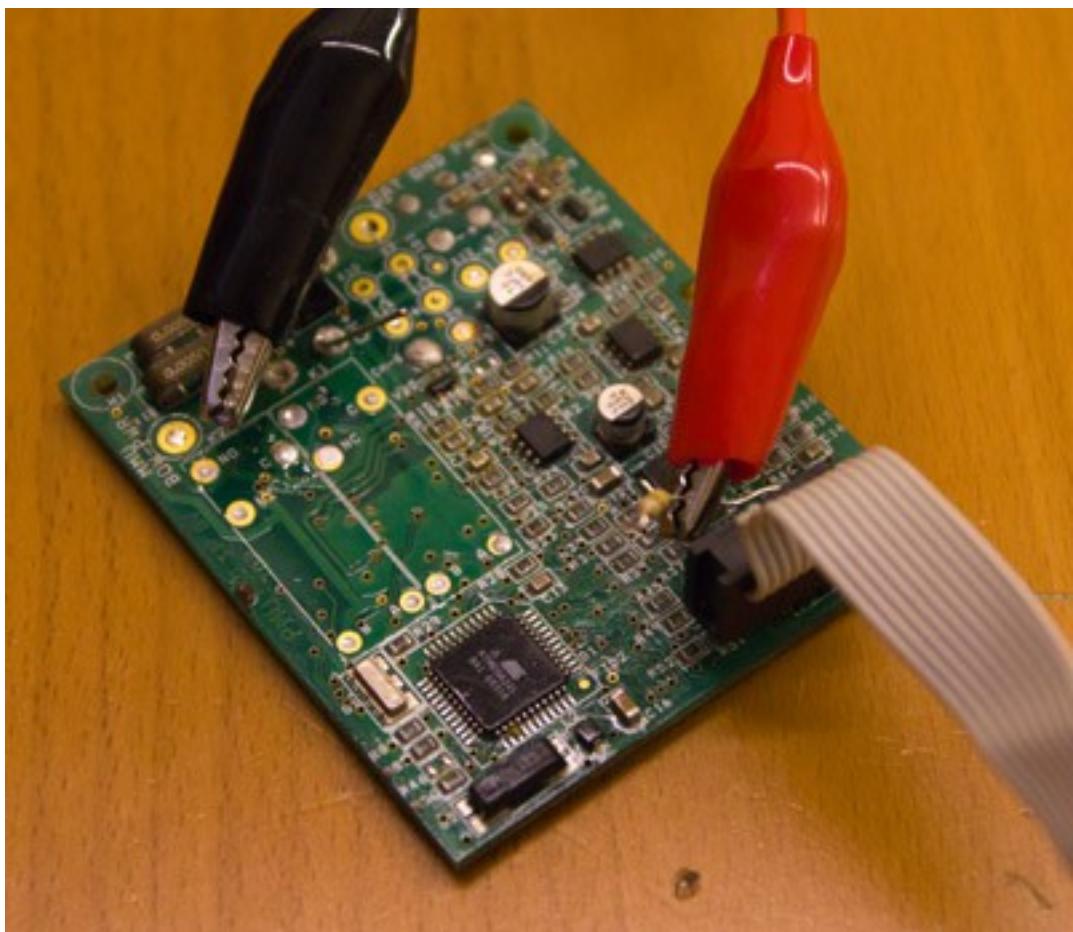
Als je een ongesorteerde doos elektronica onderdelen zou weergeven in schema vorm, zou het er ook ongeveer zou uit zien.

Het fijne aan grote SMD onderdelen als deze (de weerstanden zijn 2 bij 1.25mm) is dat je er nog gemakkelijk draadjes aan kunt solderen. Nog handiger is dat in plaats van een idiote kleur codering, de waarde er gewoon met getallen op geschreven staat. Dus dat werk wel makkelijk.

Alle kleine zwarte onderdelen kun je niet altijd iets zinnigs over zeggen, helemaal niet als het potentieel stuk is. Een drie pootje kan een spannings regelaar, diode, mosfet of transistor zijn, en de waarde mag je raden want die staat er niet op.

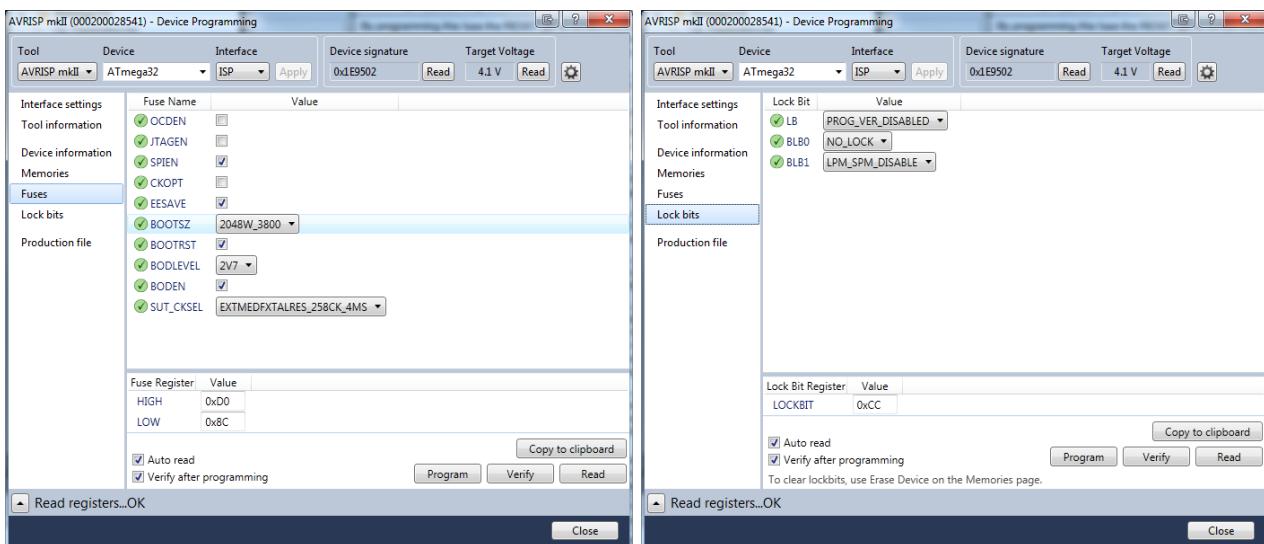
Het goede nieuws is dat de 6-pins connector direct naar de programmeer pinnen van de Atmega gaat (door een 1k serie weerstand), en de voeding door een 10 Ohm weerstand op de Vcc pin aankomt. De pin-layout is niet standaard, maar een klein verloopje zit zo in elkaar.

Omdat het printje zelf geen spanning maakt, geef ik hem extern een beetje prik. Rond de 4 Volt ofzo zou moeten werken om in ieder geval met de Atmel te babbelen.



Het is in de programmeer omgeving 1 keer raak: Hij wordt herkend! Jeej!

Daar houd het feestje helaas ook op. Niet geheel tegen de verwachting in zijn de lock bits ingeschakeld, wat betekend dat de applicatie niet uit het geheugen gelezen kan worden.



De chip start op in zogenaamde bootloader mode. (De BOOTRST fuse zorgt daar voor.) De lock bits staan zo ingesteld dat hij nooit de bootloader mag overschrijven, maar wel de applicatie. Als er niks ge-update hoeft te worden, wat bij normaal gebruik het geval is als je de fiets aan zeg, start hij de applicatie... en gaat hij banaan-management doen.

Deze lock-bits zijn dingen die op de chip letterlijk weg branden. Op deze wijze kunnen vervelende mensen zoals ik niet eenvoudig de applicatie uitlezen, maar kan een Sparta dealer wel de firmware in de accu updaten om er meer dubieuze en ongedocumenteerde functies in te prakken.

Op zich had iemand op het circuits online forum dit ook al bevestigd, 4 jaar geleden. Als je de chip wist, kun je wel het EEPROM uitlezen, en die inhoud heeft hij ook online gekwakt. Ik schiet daar nog niet zo veel mee op, dus ik ga deze niet wissen en laat hem verder met rust.

Hallo?!

Alle onderdelen in deze fiets praten met elkaar over een enkel geel draad, een bus. Deze staat normaal 24V op, en kan door een van de onderdelen (de motor, het display of de accu) gebruikt worden om over te communiceren.

Er is verder echt extreem weinig nuttigs te vinden over de elektronica in deze fiets, de enige hint komt wederom van CO, en die is dat er standaard USART op 9600 8N1 over die draad heen gaat.

De Atmega die nog met 4V aan staat, geeft op de TX pin inderdaad signaal uit:
10 04 20 CC en af en toe 10 C1 21 22 80 5F.

In banaan-taal zal dat hoogstwaarschijnlijk het volgende betekenen:

"Hello moto?"

"Hello display?"

(Of in andere volgorde.)

Op de ontvang pin verschijnt niks. De 24V bus wiebelt ook niet mee. Omdat de bus maar 1 enkel draad is om over te babbelen, kan deze print niet verzenden en ontvangen tegelijk. Sterker nog, als hij iets zegt op de bus hoort hij datzelfde bericht ook weer terug. Dat is erg handig ter controle. Als de bus namelijk kapot is, zoals nu, kun je iets zeggen:
"Echo!"

En als er dan niks terug komt... weet je dat er iets niet goed is.

De communicatie vanuit het oogpunt van de microcontroller zou er normaliter (vertaald vanuit het banaans) ongeveer zo uit kunnen zien:

"Display?"

"Display?"

...

"Ja, wat is er?"

"Niks."

"Niks."

...

"Okee."

"Motor?"

"Motor?"

...

"Ja wat moet je?"

Als je geen dubbel bericht, of geen enkel antwoord terug krijgt... is er duidelijk iets mis.

In het display

De kassa forum leden klagen dat elk wissewasje een bezoekje aan de dealer vereist, die alles aan elkaar moet aanmelden via het internet.

Dit zou in houden, dat stel ik heb een werkende fiets, ik pak een willekeurige accu van Marktplaats af en prik hem erin, dan zegt de fiets: "Dat is niet mij accu, vriend." En vertikt het verder.

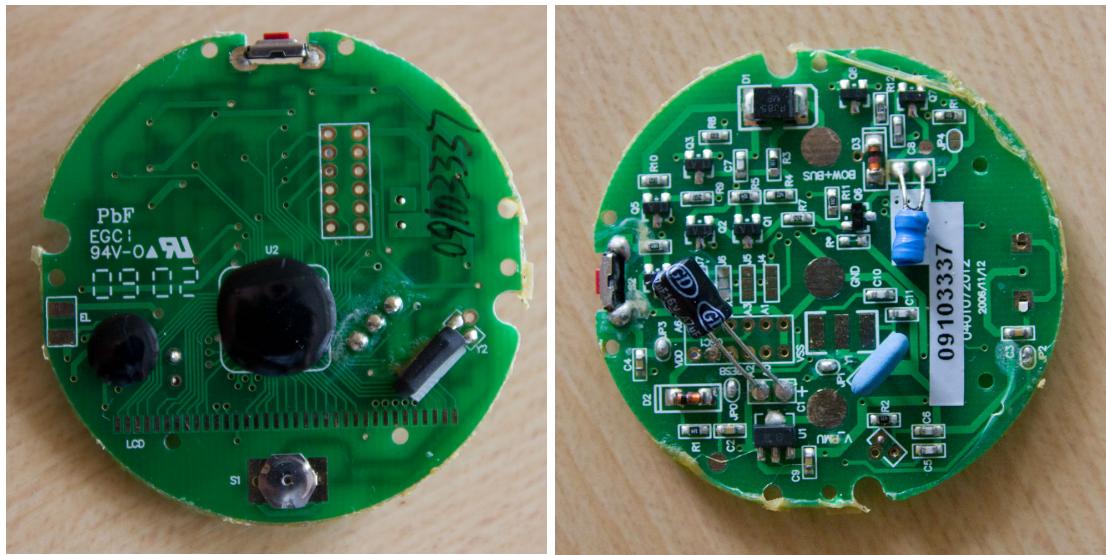
Verder wordt de suggestie op het CO forum gewekt dat er een of ander obscuur en horrific protocol overheen loopt, wat voorzien is van 1000 lagen encryptie, server communicatie,

backdoors en andere ellende die geen enkele vorm van zinnige analyse toe staan.

Ik kan het me haast niet voorstellen. Deze accu heeft al vrij veel functies, hij moet een hele berg aan metingen doen, allerlei klant specifieke dingen bij houden, zichzelf kunnen updaten, en alles door babbelen over een enkel draad.

Dat is best een complexe applicatie. Dit ding heeft maar maar 32kB aan ruimte. (De kans bestaat dat het maar 16 kB is, als je eenvoudiger een applicatie update wilt kunnen doen.) Stel dat je daar dat ook nog encryptie overheen wilt doen, moet je sleutels gaan zitten uit wisselen met de rest van de onderdelen.... het kan, maar ik schat de kans klein in.

Het display, wat ik vorige keer nog niet open had gemaakt, heeft verrassend weinig onderdelen:



Aan de ene zijde zit een bosje passieve onderdelen, en een condensator die iemand er later tussen gevrot heeft. Aan de andere kant zitten de contactjes naar het LCD, en

twee COBs.

Bijna elk LCD display wat je open maakt, in welk apparaat dan ook, heeft zo'n blob er op zitten. In 95% van de gevallen gaat het dan om een ASIC waar geen standaard verpakking omheen zit. Het chipje zit dan direct aan het PCB verbonden. Het kan ook dat ze er epoxy overheen gedaan hebben, gewoon om het je lastig te maken.

Het zou kunnen dat er een microcontroller onder de blob zit, maar ik verwacht het eigenlijk niet. Gezien er knopjes op het display zitten, en je de hele fiets met het display bedient, moet dit ook data terug kunnen sturen.

Er zit verder een stikker op, met een nummer, en met de hand is op de andere zijde... hetzelfde nummer geschreven. Als er dingen op elkaar aangemeld moeten worden, zou dat wel eens voorbij kunnen komen?

Er is meer!

Ik heb nog een hele doos met bananen staan, waar nog niks mee gedaan is.

Wellicht is het nu wel interessant om te noemen dat deze bananen ongeveer 185 kCal aan energie bevatten. ($10^2 \cdot 24 \cdot 3600 / 4184$). En dat is ongeveer 4x zoveel als een echte banaan. 

Alles is uiteraard ook verkocht als defect, dus ik verwacht er niet veel van. Maar de tweede uit de doos is wel het eerste exemplaar waar ik spanning op de display pinnen meet: 5V en 24V.

...  ...

Nou, displaytje erin prikken op hoop van zegen dan maar.

En wederom zit het mee: Het display gaat aan! (Ik dacht eigenlijk dat die ook stuk was.) Een relais in de accu zegt tik, en de display geeft een foutcode: E0005.

Als ergens veel documentatie over te vinden is in dit project, dan zijn het wel de foutcodes. De eerste online bron heeft dit over de foutcode te zeggen:

E05: Slechte verbinding naar de motor.

No shit.

Voor mij is dit veel belovend, want zo'n melding betekent data verkeer. En data verkeer, dient gesnoven te worden... met een snuiver...

Bananen taal

Door Infant op zaterdag 14 juni 2014 18:44 - Reacties (29)

Categorieën: Gepruts, Sparta: Fiets verbouwen tot het stuk gaat., Views: 16.349

Dit is in vervolg op de vorige posts:

[Dooie Fiets](#)

[Banana?](#)

Je zit nu hier:

[Bananen taal](#)

En het gaat hier verder:

[Cyclic Banana Checks](#)

[Alles moet draaien](#)

[Alles moet \(kapot\) draaien](#)

Deze post zal spelfouten, semi-technisch gebrabbel en eindeloze ritsen aan compleet nutteloze data bevatten. Als je daar allemaal geen zin in hebt, zou ik vooral niet verder lezen.

Ik was zover dat ik een accu en display op elkaar aansloot, en hij spontaan beeld gaf.

Geweldig.

De eerst volgende stap was om er een scope aan te hangen, om een idee te krijgen wat er zo ongeveer gebeurt.

Snuf snuf

Er komen 3 draadjes uit het display zetten, de accu zet daar 5V, 24V en GND op.

De 24V lijn zou de data bus moeten voor stellen, dus dat is het eerste waar de scope wat aan mag ruiken.

Er komt gigantisch veel verkeer voorbij. Er zit wel wat structuur in, bijvoorbeeld:

Elke overgang van geen data, naar wel data, begint met het karakter 0x10.

Uit de vorige post:

*De Atmega die nog met 4V aan staat, geeft op de TX pin inderdaad signaal uit:
10 04 20 CC en af en toe 10 C1 21 22 80 5F.*

Mooi. Het is nu toch wel 99% zeker, dat het standaard seriële communicatie is. Deze scope kan dat ook decoderen, maar om dat vervolgens allemaal met de hand te gaan zitten over krijten... daar heb ik niet zo'n zin in. Dat moet gemakkelijker kunnen.

Het eerste geschikte object wat ik op mijn plank tegen kom, is een USB naar RS-485 adapter.

RS-485 is een standaard, verzonden in een donker verleden door een of ander instituut.

Om bitjes te kunnen lezen en zenden, heeft RS-485 twee signalen nodig die het tegenovergestelde van elkaar zijn. Als het ene signaal hoog is, en de andere laag, komt er een 1 uit zetten, en als de andere hoog is... en de ene laag, komt er een 0 uit zetten.
Okee?

Die "andere" en "die ene" heten officiëler A en B, en iedereen (lees: Infant) haalt ze altijd door de war:

The RS-485 signaling specification states that signal A is the inverting or '-' pin and signal B is the non-inverting or '+' pin.

This is in conflict with the A/B naming used by a number of differential transceiver manufacturers...

These manufacturers are incorrect, but their practice is in widespread use.

To avoid these confusions, some equipment manufacturers have created a third D+ and

D-naming convention.

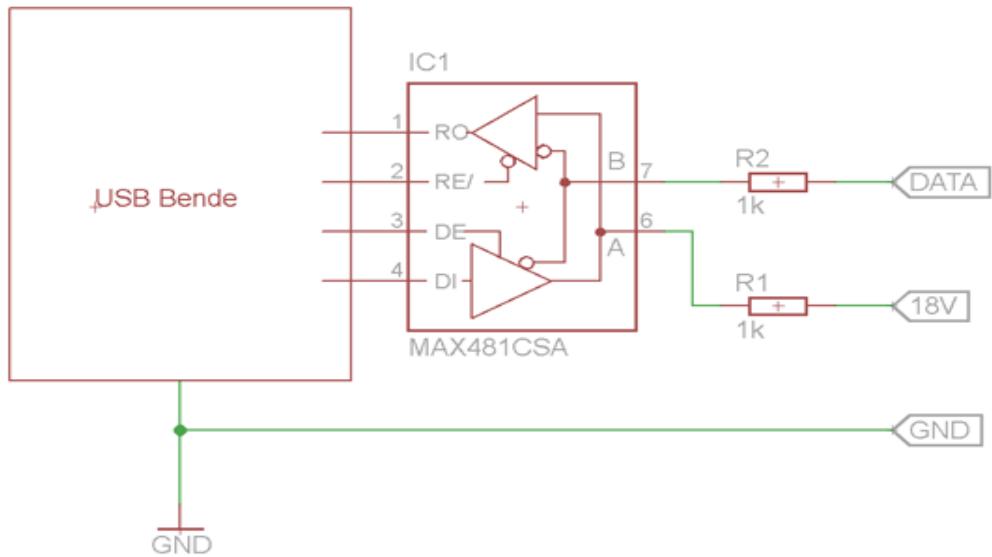
-Wiki

Okee, rant time. Kort samengevat:

De helft van de fabrikanten implementeert de standaard verkeerd. Dat vinden we met z'n allen dan onduidelijk. Vervolgens fixen we dat door de onduidelijkheid te voorzien van een nieuw label, die niet in de standaard genoemd is.

Dit is waarom elke handleiding die ik heb gelezen m.b.t. RS-485 ongeveer zo gaat:
"Sluit A op A aan, B op B. Doet ie ut niet? Draai ze om."

Anyhow, met een voeding en twee weerstanden, kan deze converter ervan overtuigd worden dat de 24V fiets bus een hele brakke RS-485 lijn is:



Na dat zo aan te hebben gesloten, en de A en B een keer te hebben verwisseld, slinger ik mijn favoriete serial terminal aan en steek het display opnieuw in de accu.

Meteen wordt je overspoeld met een muur aan gegevens, dat gaat zo'n 10 seconden door. Het display toont de foutcode E0005 net als de vorige keer, en daarna gaat het hele zaakje uit.

code:

1	10 c1 21 22 06 5d 10 22 c2 22 00 ad 77 10 a4 20 .Á! ".]. ."Â".w.€
2	ee 10 a4 20 ee 10 a4 20 ee 10 a4 20 ee 10 c1 21 î.€ î.€ î.€ î.Á!

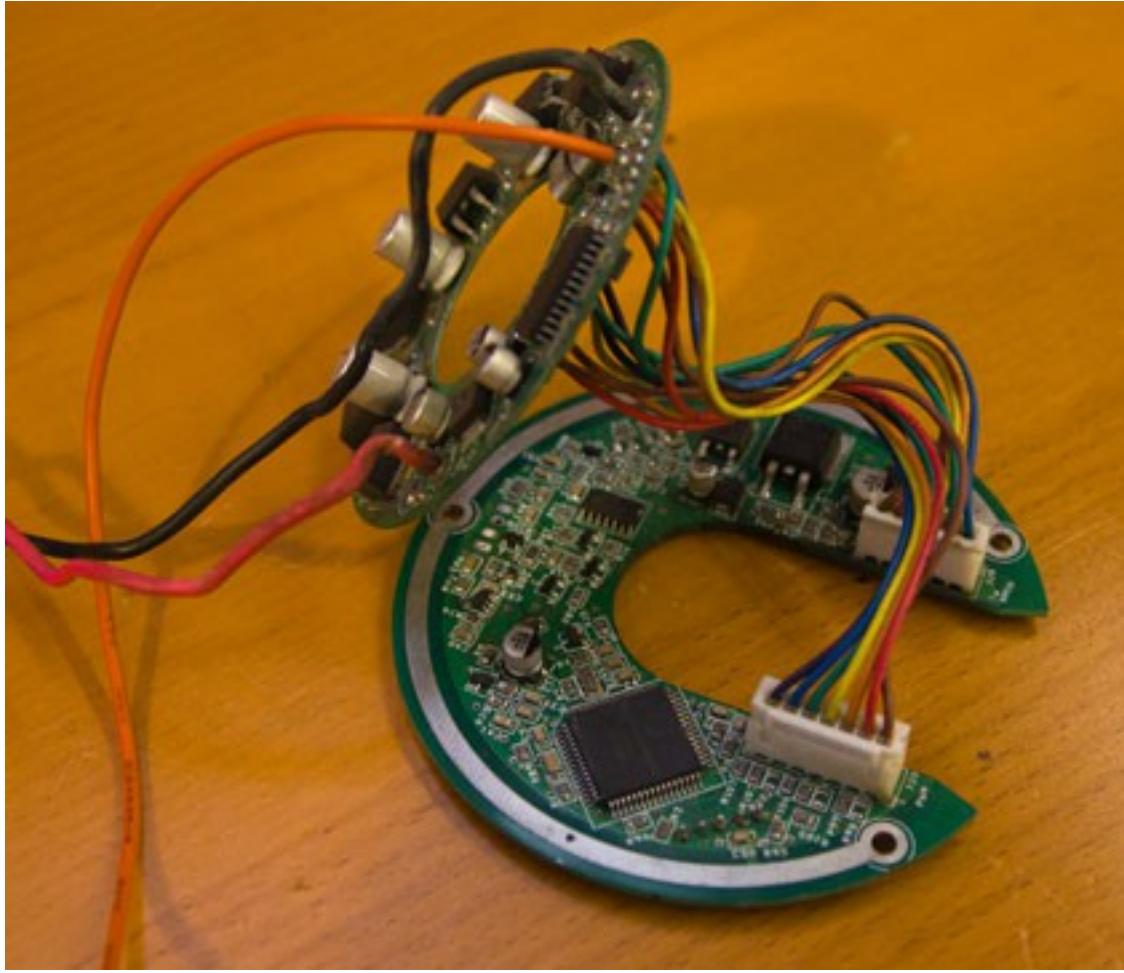
3	22 07 cc 10 22 c2 22 00 bc 2b 10 c1 21 22 08 39 ".Ì."Â".Œ+.Á!" .9
4	10 22 c2 22 00 bd ba 10 c1 21 22 09 a8 10 22 c2 ."Â".œ°.Á!" .š."Â
5	22 00 bf db 10 c1 21 22 0a 58 10 22 c2 22 00 c0 ".¿Û.Á!" .X."Â".À
6	cb 10 04 20 cc 10 04 20 cc 10 04 20 cc 10 04 20 Ë.. Ì.. Ì.. Ì..
7	cc 10 c1 21 22 0b c9 10 22 c2 22 00 d0 06 10 c1 Ì.Á!" .É."Â".Ð..Á
8	21 22 0c fb 10 22 c2 22 00 d1 97 10 c1 21 22
	0d !" .û."Â".Ñ— ; Á!" .

Kansloos!

Dit is een stukje uit dat gelogde verkeer, om je een idee te geven. Het hele bestand staat [hier](#).

Zoals je kunt zien, komt er geen enkel nuttig leesbaar stukje tekst in voor. Niet dat ik dit echt verwachtte.

Stap twee was dan toch maar eens de motor er op aansluiten, want daar zal hij toch ook wel mee willen babbelen? De motor heb ik helaas uit elkaar gehaald, maar de ingewanden zijn nog heel:



Dus, die draadjes vrot ik in de daarvoor bestemde connector, start een nieuwe log sessie (dataverkeer wat er bij hoort.), en wat schetst de verbazing?



OMFSM! Een getal!

Het eerste wat je dan natuurlijk op stom geluk doet, is kijken of dat getal ergens toevallig voorbij komt vliegen.

Ik heb de data in het log in regels opgesplitst, zodat op elke regel een begin van een pakketje staat.

Meerdere malen komt 10 C1 29 26 0c 0c c3 54 c0 00 f0 59 41 21 voorbij, en dat is het enige waar 59 41 zo pats boem in voor komt. (De hex waarde van 59 41 is 17 35 of 10 3F, afhankelijk van de endianness, maar beide komen niet voor.)

Dat is vrij veel belovend, maar nog niet echt een zekerheid.

De laatste banaan die ook nog een teken van leven toonde, heb ik er toen ook maar aan gehangen en gelogd.

Deze doet het ook, en zodra het display de km stand geeft haal ik hem er weer uit. Dit

display geeft 7052 km aan, en what do you know: vlak bij het einde van het log staat:

10 C1 29 26 03 0c c0 00 c0 00 f0 70 52 49

Consistentie!

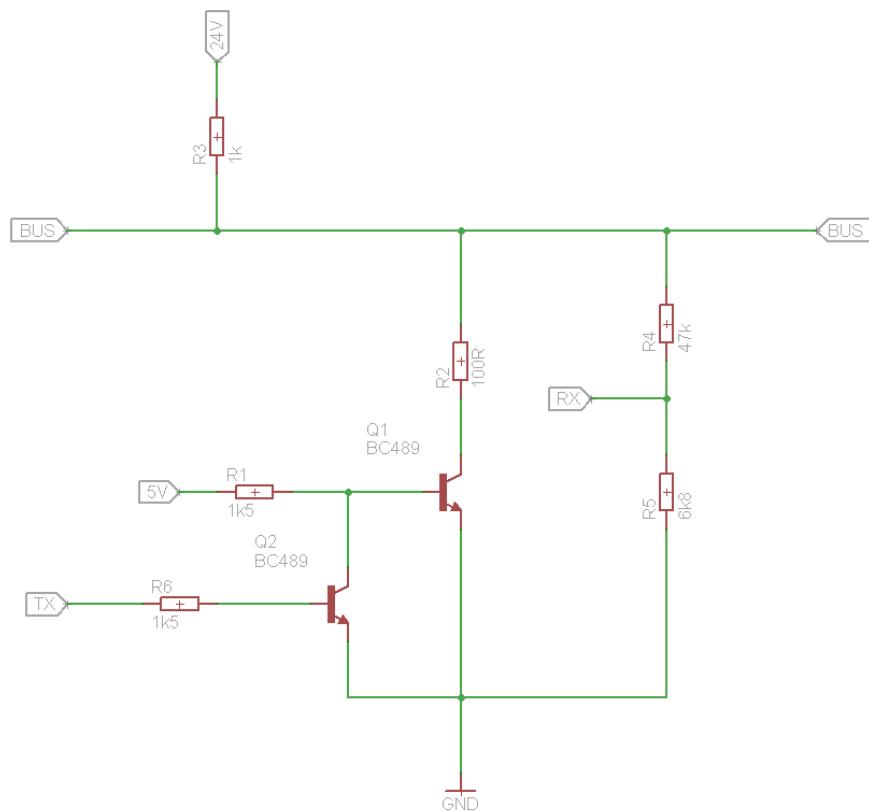
Veel meer kan ik van deze opstelling natuurlijk niet maken. De "motor" gaat nooit kunnen draaien, het is überhaupt een wonder dat hier leven uit komt.

De volgende stap leek mij daarom, uitvinden of ik zelf iets tegen het display kan zeggen, bijv een andere kilometer stand. Daarvoor moet ook terug gepraat kunnen worden, en dat vereist een iets andere en wellicht ook nettere opstelling.

Terug babbelen

In plaats van het gebeunde RS-485 ding nog langer te misbruiken, heb ik een oud Olimex bordje uit de kast gehaald.

Deze heeft er een RS-232 geval erop zitten, wat weer zo'n standaard is die gewoon hetzelfde op een andere manier doet, en daar heb ik het volgende op geprikt:



De communicatie bus, is een los draad wat d.m.v. een pull-up weerstand(R3) op 24V gehouden wordt. Als de bus 24V is, maakt de deling R4/R5 daar ietse van 8 keer minder van (3V ongeveer) zodat dat op het ontvangst pinnetje van het RS232 ic kan zonder dat die direct in rook op gaat.

Om de bus omlaag te trekken, moet het 5V tx-signal ge-level-shift worden naar 24V. Dat kan met twee transistoren: Als tx hoog is en Q2 in gaat, is de ingang van Q1 laag en laat hij de bus met rust. De bus is dus 24V.

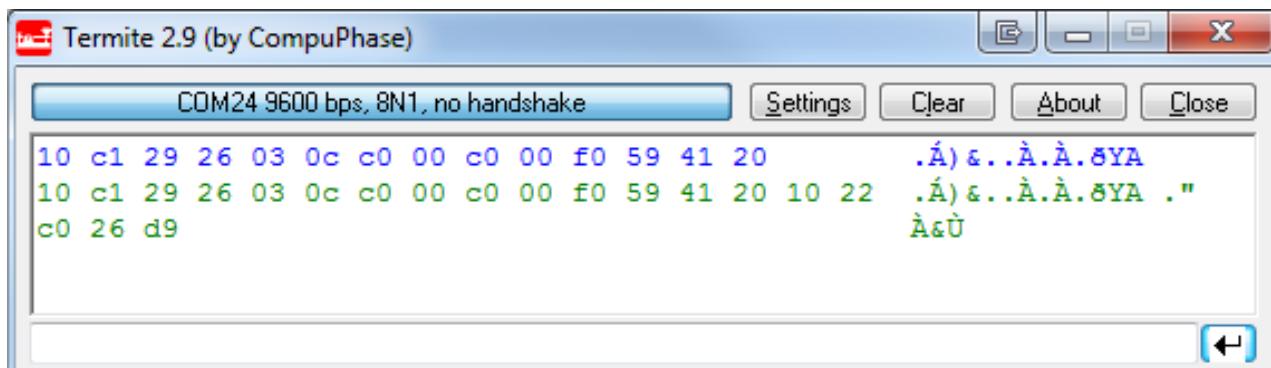
Als tx laag is, wordt de ingang van Q1 hoog, en trekt hij de bus door een 100 Ohm weerstand R2 naar de GND toe. De bus is dan nog ongeveer 2V, gedeeld door c.a. 8 is dat laag genoeg om als 0 gelezen te worden.

Een nuttige feature die deze bus automatisch heeft, is dat alles wat je nu over RS-232 verzend, ook direct terug krijgt. Als twee dingen op exact hetzelfde moment op deze bus proberen te praten, kun je dat direct vast stellen, je krijgt namelijk niet hetzelfde terug als wat je er op zet.

Het bordje ziet er nu zo uit. Er zit een connector op waar het display in geprikt kan worden. En als je dat doet, gebeurt er niks.

Hm...

Misschien moet hij eerst wat aan gemasseerd worden, door er data naar toe te slingeren. Dus ik verzend het stuk waarvan ik vermoed dat het 5941 km op het display weer gaat geven:



Ooh!

Voor een halve seconde ofzo verschijnt inderdaad dezelfde km stand op het display, en verdwijnt daarna weer. Als ik eerst dit bericht, en daarna allemaal onzin verzend blijft het

display aan, net zolang tot ik ophoud met verzenden.

Veel interessanter nog, is dat het display kennelijk antwoord geeft:
10 22 c0 26 d9.

Het versturen van het pakket met de andere km stand, geeft hetzelfde antwoord.

Dat is heel aardig, want dan kan ik wat van het gelogde verkeer op een rijtje zetten:

```
10 C1 29 26 0C 30 C3 54 C0 00 FC CC C0 71  
10 22 D0 26 D9  
10 C1 29 26 03 0C C0 00 C0 00 F0 70 52 49  
10 22 C0 26 D9  
10 C1 29 26 0C 30 C3 4F C0 00 FC CC C0 4E  
10 22 C0 26 D9
```

Die gaven allemaal hetzelfde antwoord, maar er is ook:

```
10 C1 29 27 03 30 C0 00 00 00 3C CC C0 D4  
10 22 C0 27 48  
10 C1 29 27 03 00 00 00 00 00 1e 00 05 b7  
10 22 c0 27 48
```

Bij tet verzenden van deze twee pakketjes, gaat het display niet meer uit. En kijk:

```
10 C1 29 27 03 00 00 00 00 00 1e 00 05 b7  
10 22 c0 27 48
```

Deze geeft de foutcode E0005 permanent weer!

Verder komt dit bericht vaak voorbij:

10 C1 21 22 00 FE.

Het een na laatste getal loopt steeds op:

```
10 C1 21 22 01 6F  
... 02 9F  
03 0E  
04 3C  
05 AD  
06 5D  
07 CC  
08 39  
09 A8  
0A 58
```

0B C9
0C FB
0D 6A
0E 9A
0F 0B

En na 0F begint hij weer opnieuw.

Dit doet sterk vermoeden dat de laatste byte een CRC is. Als ik namelijk een bit aanpas in het bericht naar het display toe, en bijv. foutcode 0004 ervan maak, gebeurt er niks, en krijg ik ook geen antwoord.

En ik wil zo graag dat het display 666km aangeeft.

CRCs

CRC's zijn geweldige dingen. Het is een controle getal wat in dit geval achter het bericht geplakt zit.

Als tijdens de verzending de data in het bericht veranderd, komt de CRC niet meer overeen. Ze zijn ontworpen om bij een kleine verandering, van. bijv maar 1 bit, een zo groot mogelijke verandering in het controle getal op te leveren.

Maar een 8-bit CRC is natuurlijk niet zo veel. 8 bits zijn 256 verschillende mogelijkheden.

Als ik zo'n bericht compleet omgooi, en er een willekeurige CRC achter bengel, heb ik een 1 op 256 kans dat ik die goed gok.

En gezien dit display zo vriendelijk is om niks te zeggen als ik puin verstuur, en netjes te antwoorden als hij het met me eens is, kan ik in theorie gewoon maar wat getallen op het eind proberen totdat het goed gaat.

Nou, dat gehele proces heb ik samengevat in een quick and dirty programmaatje.

*(Mocht je het willen proberen... het werkt onder Linux of Windows + cygwin door make uit te voeren. Hij opent standaard /dev/ttyS23, maar je kunt een ander device als opstart argument mee geven. Bijv:
code:*

```
1 ./tryme /dev/ttyUSB0
```

Als je die opstart, vraagt hij weke waarde hij naar het display moet smijten:

```
$ ./tryme
Uart /dev/ttyS23 is opened
Uart /dev/ttyS23 opened at rate [9600]
Infant's Useless Display-O-Matic.

Hello user. How fast do you want to go?
666
```

Het getal wat je invoert gaat door een binare gehaktmolen heen:

C:

```
1 //The encoding is kinda weird, it's the HEX representation of what you
2 uint8_t dec[3];
3 //Break it into powers of 10.
4 dec[0] = speed / 100;
5 dec[1] = (speed - (100*dec[0])) / 10;
6 dec[2] = speed % 10;
7 //Convert it to ...something
8 data[8] = 0x01 * dec[0];
9 data[9] = ((0x10 * dec[1])) | dec[2];
```

Dat propt hij op de plek waar ik ondertussen stiekem van heb uitgevonden dat het de speedo locatie is, hij laat zien wat hij verzonden heeft en wat het vorige antwoord was:

```
Sending: 10 C1 29 26 03 0C C0 00 06 66 F0 00 00 27
Response: 10 C1 29 26 03 0C C0 00 06 66 F0 00 00 26
```

En na een aantal pogingen komt er een antwoord terug van het display:

```
Sending: 10 C1 29 26 03 0C C0 00 06 66 F0 00 00 C4
Response: 10 C1 29 26 03 0C C0 00 06 66 F0 00 00 C4 10 22 C0 26 D9
```

En verschijnt het getal automatisch op het scherm!



Enig.

We zijn nu in ieder geval op het punt beland dat je Sparta Ion display in combinatie met deze enigszins lompe manier van communiceren, kan omtoveren tot bijv een kookwekker.

Hij piept alleen niet...

Cyclic Banana Checks

Door Infant op woensdag 30 juli 2014 18:15 - Reacties (15)

Categorieën: Gepruts, Sparta: Fiets verbouwen tot het stuk gaat., Views: 9.930

Dit is een vervolg op:

[Dooie Fiets](#)

[Banana?](#)

[Bananen Taal](#)

Het gaat zeker helpen om die een beetje door te lezen, anders heb je geen flauw idee waar dit over gaat.

Mijn titel-verzin-o-matic kwam na afloop met deze zeer gepaste titel: Cyclic Banana Checks.

Dit gaat waarschijnlijk ook de meest on-duidelijk post tot nu toe worden, maar ja... het dient voornamelijk ter documentatie.

Kuch pas op spel fouten kuch

De vorige post hield op met de cliffhanger dat er wellicht nu een kookwekker van de ronde Batavus display gemaakt kan worden. Dat gaan we natuurlijk niet doen, om meerde redenen:

1: Ik hoef geen kookwekker.

2: Hij piept / vibreert niet, wat een must is voor een degelijke kookwekkert. (geen typo)

En als derde punt: Mijn PC is c.a. 50W aan vermogen aan het verstoken, om er data naartoe te slingeren. 50W zou je, mits hij het aan kan, toch nog 3 minuten lang uit een AA batterij kunnen halen.

(En als hetgene wat je aan het kook-wekken bent bijv. een zacht gekookt ei moet worden, kan die 50W voor ei-verwarming gebruikt worden.)

Dat is allemaal vrij aardig, maar zo gezegd sub-optimaal.

Hoe kan het optimaler?

Het probleem is nu, dat we wel berichtjes naar het display kunnen sturen, maar we moeten steeds de checksum aan het einde gokken. Omdat de communicatie ook nog eens moeilijk traag is, staat de PC daar eigenlijk 99% van de tijd op te wachten.

Om de volgende onorthodoxe en enigszins lompe manier van aanpak te volgen, is het handig dat je een soort van een idee hebt hoe een crc ongeveer werkt. In plaats van de [Wiki](#), kan ik deze aanraden:

[A Painless Gguide To CRC Error Detection](#)

Op zich hoef je helemaal niet te weten hoe een crc precies werkt om ze te kunnen gebruiken. Er staan namelijk talloze voorbeelden online, die al in software verwerkt zitten. Pak bijv deze 8-bit CRC uit een of andere linux kernel module:
<http://lxr.free-electrons.com/source/lib/crc8.c>

Of deze bijv. bijv uit [chromium](#):

De laatste gebruikt een tabel waarmee de CRC berekeningen gedaan worden, de eerste niet.

Met een tabel-loze functie, zou je zo'n tabel kunnen genereren. Dat is wat reken intensiever. Maar daarna kun je de snelle tabel-functie kunt gebruiken, met als nadeel dat die wat meer geheugen gebruikt, en maar een vaste CRC variatie kan implementeren.

Een andere eigenschap van een CRC functie kan zijn dat je hem in een of meerdere stappen kan uitvoeren:

C:

```
1
2
3 //Any CRC8 function interface:
4 uint8_t crc8(uint8_t* data, uint8_t len, uint8_t crc);
5
6 //Data
7 uint8_t data[4] = {1,2,3,4};
8 uint8_t crc;
9
10 //In one go:
11 crc = crc8(&data[0], 4, 0x00);
12
13 //Same result, in two goes:
14 crc = crc8(&data[0], 2, 0x00);
15 crc = crc8(&data[2], 2, crc);
16
17
```

Hierboven een stukje voorbeeld code:

Als eerste een definitie naar een willekeurige crc8 functie. Deze functie geeft als output een crc getal. Als input moet je hem een stuk data geven, en de lengte waarover de crc berekend moet worden.

Ik zou de functie in twee stappen kunnen aanroepen, of in een enkele keer. Als ik het in twee stappen doe, geeft ik de crc die de eerste stap oplevert mee als argument aan de

tweede. In beide gevallen is het argument in de eerste stap 0x00, en het eindresultaat hetzelfde.

Je zou dit in zoveel stappen kunnen doen als je zelf wilt. In de code uit de eerste link is dat het makkelijkst te zien:

C:

```
1 while (nbytes-- > 0)
2     crc = table[(crc ^ *pdata++) & 0xff];
3 return crc;
```

Als kan deze loop halverwege afbreken, en in een nieuwe verder gaan met dezelfde crc.

Stel dat we zo'n functie, waarvan we de details nog niet precies weten, op de volgende display data willen los laten:

10 C1 29 26 03 0c c0 00 c0 00 f0 13 37 68

Dit gaat overigens 1337 op het display zetten:



Het probleem is dat ik niet weet waar de crc precies over berekend wordt. Het kan over alles zijn, het kan dat bijv de eerste waarde C1 een samenvoeging van 0xC0 en 0x01, en

dat daar een CRC over gedaan wordt. Maar het wordt in ieder geval over de data gedaan, en die staat aan het einde.

Dus ik dacht: Als ik nou een tabel maak, waarin ik voor verschillende getallen aan het einde van de data, de correcte CRCs opslag, dan kan ik in ieder geval later op mijn gemak een crc functie gaan zoeken die diezelfde output produceert, zonder dat ik deze hele opstelling mee hoeft te nemen.

Nou, zo gezegd, zo gedaan. Dat levert dus weer een klein programmatie op. Dat heeft iets van drie kwartier getallen naar het display geschoten, en leverde toen de volgende tabel op:

code:

```
1 Data from 01
2 5F CE 3E AF 9D 0C FC 6D 98 09 F9 68 5A CB 3B AA
3 .....
```

De data aan het einde van het display commando was dus 0x01 0x00 en levert in dit geval CRC 0x5F op (Eerste waarde in de tabel).
0x01 0x01 leverde CRC 0xCE op, etc.

Maar er is een probleem: 0xFF komt 4 keer voor, en dat zou niet moeten. (Alles hoort in een crc tabel maar 1 keer voor te komen).

Dat zou een aantal dingen kunnen betekenen:

- 1: Dat er drie getallen zijn die nooit geaccepteerd worden door het display.
- 2: Ik heb iets fout heb gedaan.
- 3: Er wordt een brakke/rare crc gebruikt.

2 lijkt me het meest voor de hand liggende. Om het iets zekerder te weten, besloot ik om er ook maar bij te loggen of het antwoord terug ontvangen ook gelukt is ja of nee, en laat dat een nachtje draaien.

De volgende dag had ik een berg met prachtige tabbelen, behalve 1tje.

code:

```
1 Data from 10
2 FF FF
3 .....
```

4	Succes table:
5	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Moment dat ik ergens in mijn fop-data een 0x10 stop, faalt het en is het bericht nooit goed.

Nou... dat zou op zich kunnen, en zou betekenen dat je het nooit kan zien als je exact 10 km/h fietst. Maar hij zou dan ook nooit kunnen weergeven dat je 10km hebt afgelegd.... niet erg waarschijnlijk.

Het bericht begint ook altijd met 0x10... dat is vast geen toeval. Grote kans dat er iets van escaping wordt toegepast.

Zucht, dat kan natuurlijk ook op 100 verschillende manieren, maar ik dacht, ik probeer het door achter elke 0x10 in de data gewoon nog een 0x10 te vrotten.

...en dat werkte in één keer! Feest!

En nu?

Het volgende programmatje heb ik gewoon een hele berg crc functies in gepropt, en die gaat dan domweg wat eigenschappen naar de crc functie slingeren, en een tabbelletje bijhouden, net zolang tot die overeen komt met de hiervoor ge-log-te tabellen. (Ge-logde? het krijgt allebij geen rode streepies!)

```
Using CRC8_2:  
Highest number of matches so far:  
INIT: 0x17  
FINAL: 0x00  
POLY: 0x0000  
MATCHES: 3  
Currently doing:  
INIT: 0xA0  
FINAL: 0x86  
POLY: 0x0001  
MATCHES: 0
```

De init waarde is het argument wat aan de crc functie mee gegeven word. Omdat de tabellen die ik zojuist gemaakt heb halverwege de data begonnen, weet ik niet wat de vorige crc waarde was, dus daar probeert deze gewoon alle combinaties van.

Een crc wil ook nog wel een xor operatie aan het einde doen, en ten slotten kunnen

de operaties in de functie in LSB of MSB volgorde met een verschillend aantal polynomen gedaan worden.

Dit programma probeert gewoon alles, en dat zou best wel eens even kunnen duren.

Bij de tweede crc functie die hij ging proberen, kwam er na een aantal minuten al het volgende te staan:

```
Using CRC8_2:  
Highest number of matches so far:  
 INIT: 0xD4  
 FINAL: 0x00  
 POLY: 0x0042  
 MATCHES: 253  
Currently doing:  
 INIT: 0x30  
 FINAL: 0x06  
 POLY: 0x0051  
 MATCHES: 1
```

253 matches! Dat betekent gewoon dat hij het heeft gevonden!

De persoon die dit protocol verzonden heeft heeft er vast met opzet 0x42 als magisch polynoom in gestopt. Ik weet het zeker.

Nu de gebruikte crc functie bekend is, is de laatste stap uitvinden over wat de crc berekend wordt. Dat was ook een vrij kort stukje code.

Ik nam dan twee willekeurige berichtjes die gelogd waren. Als je nu de crc functie het correcte startgetal en startpositie mee geeft, en over het hele bericht **inclusief** de crc laat heen lopen, zouden beide hetzelfde getal moeten op leveren. Dat is ook een handige eigenschap van een crc functie.

Dat was in no time gevonden: Het start getal is 0x07.

De uiteindelijk crc-code is een aanpassing op code die [hier](#) vandaan komt:

C:

```
1 /*  
2  * CRC-8 for Sparta ION bus communication.  
3  */  
4 uint8_t crc8_bow( uint8_t *data_in, uint8_t len, uint8_t init)  
{
```

```
uint8_t crc;
uint8_t i,j,f;
uint8_t data;
crc = init;
for (j = 0; j != len; j++) {
    data = data_in[j];
    for(i = 8; i; i--) {
        f = ((crc ^ data) & 0x01);
        if (f == 0x01){
            crc = crc ^ 0x42;
        }
        crc = (crc >> 1) & 0x7F;
        if (f == 0x01){
            crc = crc | 0x80;
        }
        data = data >> 1;
    }
}
return crc;
```

En je dient hem als volgt aan te roepen:

C:

```
1 //Do a CRC check:  
2 uint8_t crc = crc8_bow(data, len, 0x07);
```

Het lollige is nu dat ik zo snel als het display het aan kan er berichtjes naar toe kan

schieten.

(Hij haperde hier af en toe nog omdat ik 0x10 nog niet escapde.)

[Youtube](#)

Niet alleen dat, ik kan nu het verkeer fatsoenlijk in kaart gaan brengen. Van elk bericht kan ik verifiëren of de CRC okee is, en nog belangrijker: Ik kan zelf berichten samen stellen, en kijken of daar een antwoord op komt.

Nou, hier kwam een bedroevend lage hoeveelheid elektronica en mechanisch geneuzel aan te pas, namelijk niks. Dat zal ik volgende keer goed maken.

Alles moet draaien

Door Infant op zondag 7 september 2014 19:21 - Reacties (10)

Categorieën: Gemod / Gefix, Sparta: Fiets verbouwen tot het stuk gaat., Views: 12.512

Dit is een vervolg op:

[Dooie Fiets](#)

[Banana?](#)

[Bananen Taal](#)

[Cyclic Banana Checks](#)

En je zit nu hier: [Alles moet draaien](#)

[Volgende deel](#)

Om het wrak dat heet mijn fiets in ieder geval van verlichting te voorzien, had ik een ruime tijd geleden al bedacht dat het wel lollig zou zijn als ik dat op zijn minst iets kon doen met die roterende bonk koper in mijn achterwiel, gezien het op dat moment niet zo veel deed.

Mijn ervaring met fietslampjes is dat ze altijd stuk zijn of slecht contact maken. Of het blijkt dat een of andere onverlaat de draadjes heeft gejat. Of je dynamo zit vol met zand/sneeuw/blaadjes of andere NS-smoesjes, waardoor het allemaal niet werkt.

Een motor die werkt als dynamo daarentegen, werkt tot nu toe vrij geweldig. Ik heb dat ding al meer dan een jaar als dynamo laten werken, en hij doet het nog steeds. En dat is lang, voor mijn doen. (Ik hou o.a. erg van trappetjes op en af rijden..)

Normaliter is een dynamo iets van 6V. Deze niet. Om de 0 tot veel volt die hier uit komt

vliegen tot iets zinnig om te zetten, heb ik daar voor zo'n Chinees doosje voor van eBay gesloopt, en op een heel andere manier toegepast dan normaal. Hij zat zonder accu aan gesloten. Dan maakt hij zelf c.a. 27V als je eerst voldoende snelheid maakt, en op die 27V konden dan weer mooi 2 Chinese 12V led lampen branden.

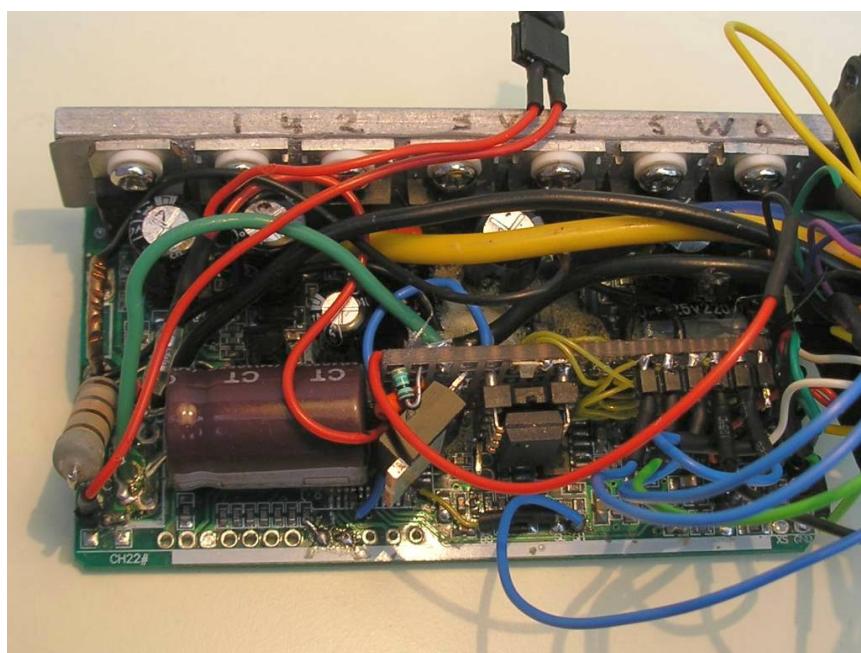
Dat werkt op zich vrij aardig, maar heeft ook een aantal nadelen:

- De verlichting staat nu altijd aan, ook overdag.
- Er komt een irritant hoge pieptoon uit mijn frame, als ik mijn hoofd 2 centimeter beweeg.

Nou valt dat allemaal nog wel op te lossen. Deze Nederlander is ook met deze controllers bezig geweest: <http://www.avdweb.nl/sola...u63-motor-controller.html>

En deze meneer heeft een vrij leesbare en gedetailleerde verzameling foto's, schema's en feiten op een rij gezet, zodat je in theorie niet eerst 32 Euro aan kapitaal hoeft te verbranden om te weten wat je in huis gaat halen.

Hij heeft er een zogenaamd legislation-device in gemaakt, wat inhoud dat het doet wat het volgens de wet moet doen: mee trappen. En dat heeft heeft hij prachtig weg gewerkt:



Ineens is het duidelijk waarom er voor een één aderig communicatie protocol is gekozen: Het maakt het leven van degene die alles in en uit elkaar moet halen een stuk eenvoudiger en vereist zo veel minder mechanisch geneuzel. (Als ik ergens moe van wordt, dan is het wel mechanisch geneuzel.)

Als ik bij deze fiets de motor los wil maken, haal ik de moertjes los. Doe ik "Plop", en voila. De motor is los.

Het is alleen zo jammer dat het vervolgens never nooit meer gaat werken omdat de dealer er eerst wat software-voodoo op los moet laten omdat Infant er weer zo nodig met zijn nieuwsgierige jatten aan moest zitten.

En dat Chineese ding gaat natuurlijk never nooit met het lollige ronde display werken waar ik jullie de afgelopen tijd mee verveeld heb. Kortom:

Tijd om zelf een motor controller in elkaar te zetten:

Hoe moet dat?

Als je zelf iets gaat maken, helpt het altijd om te kijken of niet iemand anders het stiekem ook gedaan heeft. Motoren aansturen is niets nieuws, en er bestaat dan ook een Open source motor controller.

Hier zit een STM microcontroller op die alle aanstuur logica doet.

In de orginale Sparta en Batavus motoren die ik heb, zit een Motorola chip. En voor specifiek die chip, is ook een application note beschikbaar.

Ze zijn echter nergens echt te koop, en al waren ze dat wel: ik heb er toch geen programmeer spul voor liggen, of zin om er aan te beginnen.

Ik ga deze met een ATxmega opbouwen.

Op wat merk component keuze na, zitten al deze controllers verder stiekem hetzelfde in elkaar. (Ze moeten ook hetzelfde doen... dus dat is niet heel vreemd.)

De motor die deze elektronica moet aansturen is een zogenaamde Brushless DC Motor. Het is eigenlijk een soort combinatie tussen brushed DC, een stappen motor en een 3-fase AC motor.

Een brushed DC motor gebruikt kool borstels om de stroom richting door de spoelen om te draaien, hier moet de elektronica dat gaan doen. Dit scheelt ons een setje koolborstels.

Het heeft net als een DC motor magneten in het rond draaiende deel zitten (de rotor). Het heeft drie aansluit draden. Net als de 3-fase draden uit een 3-fase AC motor.

En het heeft 8 pool paren, wat inhoud dat als de drie fases een rondje gedraaid hebben, de motor maar 1/8e rondje gemaakt heeft. (Dat is een beetje stappen motor-achtig.)

Het duidelijkste plaatje wat ik heb kunnen vinden...

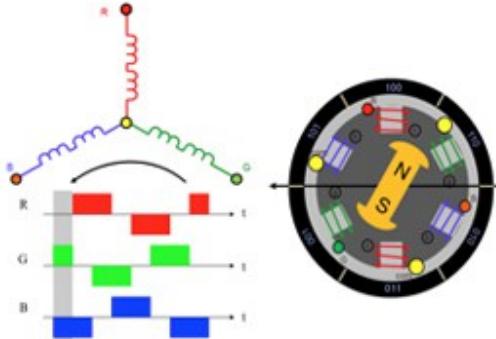


Figure 4(A): Phase 1

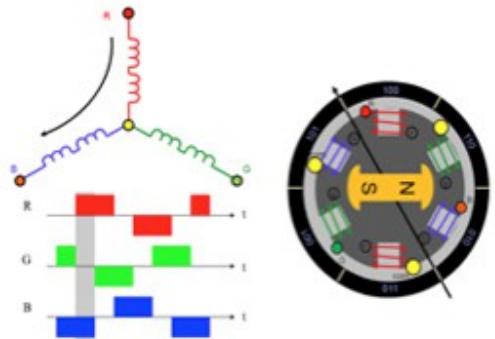


Figure 4(B): Phase 2

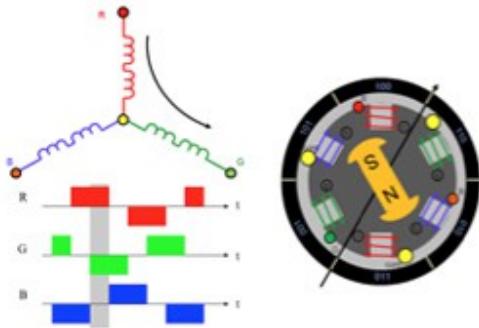


Figure 4(C): Phase 3

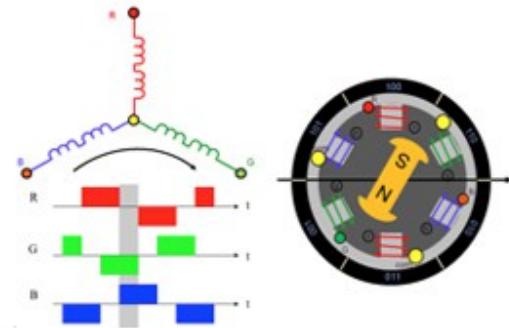


Figure 4(D): Phase 4

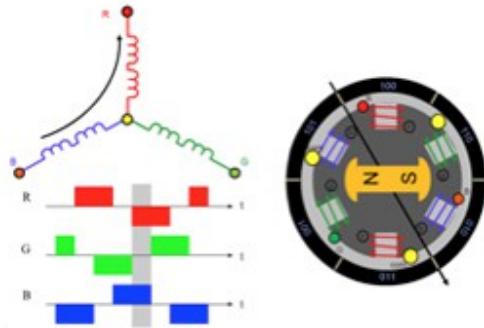


Figure 4(E): Phase 5

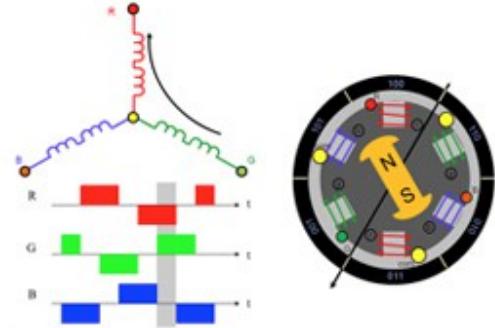
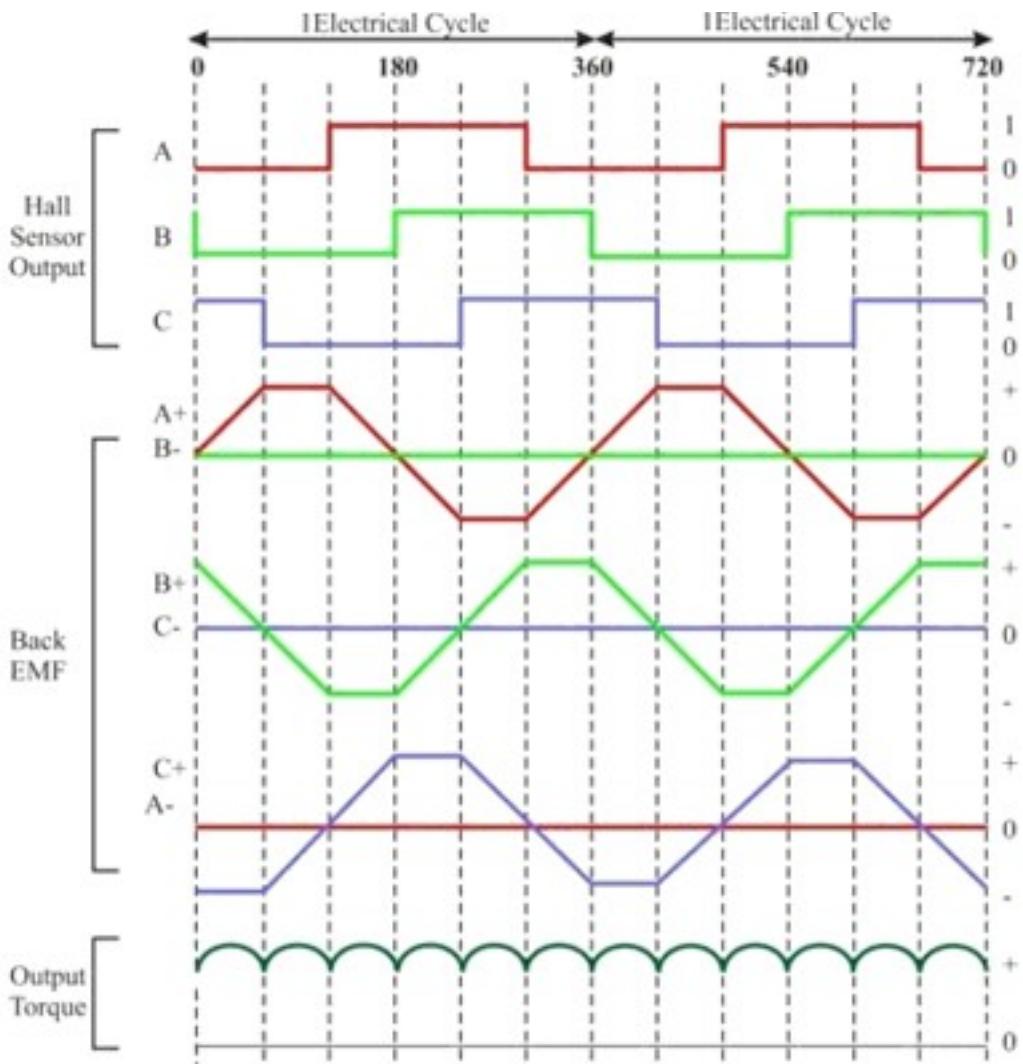


Figure 4(F): Phase 6

...laat zien dat er 6 verschillende mogelijkheden zijn waarin de stroom kan rond lopen. Als de controller dit fout doet, gebeurt er of niks, of de motor blijft vast zitten, of hij draait de verkeerde kant op.

Om het gemakkelijker goed te laten gaan, zitten er drie hall sensoren in deze motor. Deze geven aan in welke positie de magneten staan, zodat je weet welke van de 6 mogelijkheden je moet kiezen om de motor vooruit of achteruit te laten draaien.

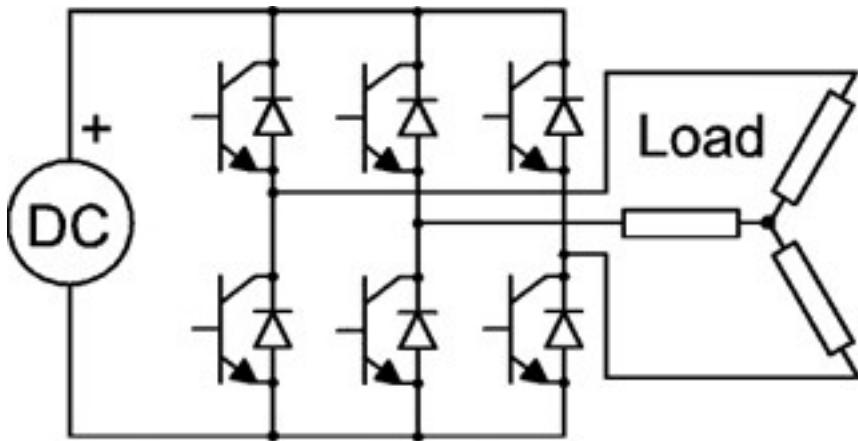


De back-EMF van deze motoren heeft een trapezoïde vorm, i.p.v. een sinusoïde.

Hall sensoren komen in veel verschillende smaken. Degene die hier in zitten zijn zogenaamde open-drain types. Dat wil zeggen dat je hun uitgang d.m.v. een pull-up op 5V moet zetten, die laten ze met rust als het signaal hoog is, en trekken ze omlaag als het signaal laag is.

De Chinese motor controller gaat van precies het omgekeerde uit (hij trekt ze zelf omlaag) en ziet als je ze direct aansluit geen sensoren. Hij kan vervolgens wel de motor laten draaien door naar de back-EMF te kijken, maar vooral bij lage snelheid loopt dit niet zo lekker.

Het belangrijkste ding wat al dit stroom-omdraai werk gaat verrichten zijn 6 mosfets gerangschikt in 3 halve bruggen:



Een synchrone buck DC-DC converter gebruikt ook een halve brug, om van een hoge ingangsspanning een lagere uitgangsspanning te maken. Hier gebeurt per fase hetzelfde. Bij een laag toerental krijg je minder spanning over een winding dan de accu spanning, hoe hoger het toerental hoe meer spanning je nodig hebt om vooruit te blijven gaan.

Een buck converter kan ook van de uitgang naar de ingang werken (door alleen maar de manier van schakelen te veranderen), en de lagere spanning van de motor terug de accu in persen. De motor gaat dan tegen werken, en als je verder niks doet, rem je dus af.

Het is verder de taak van de microcontroller te zorgen dat er niks spontaan in de fik vliegt. Als je bijv. accu en een snel rond draaiende motor hebt, en je denkt: "Kom, laat ik eens heel hard gaan afremmen." kan de stroom die het remmen oplevert nergens heen. Het gevolg is dat de spanning door het dak schiet en het gros van de componenten sloop.

Dit brengt mij tot de twee belangrijkste eisen van het eerste prototype, namelijk:

- 1: Het moet de motor kunnen laten draaien en vooral ook weer afremmen.
- 2: Het moet niet stuk gaan, ook al trek ik halverwege het testen accu's en kabels los.

Vervolgens moet het natuurlijk (niet per-sé in volgorde):

- 3: Met een rem en een gas hendel kunnen werken. (Met zo'n mumsel bijvoorbeeld.)
- 4: Moet met het display kunnen babbelen.
- 5: Moet in het wiel passen en trap ondersteuning kunnen doen.
- 6: Moet knetter hard kunnen gaan tot er dingen gaan smeulen.

De hardware

De orginele motor controller in deze motor, bestaat uit twee losse printjes.

Op de power-print zit de aansluiting die het wiel uit gaat.

De 6 mosfets + gate driver.

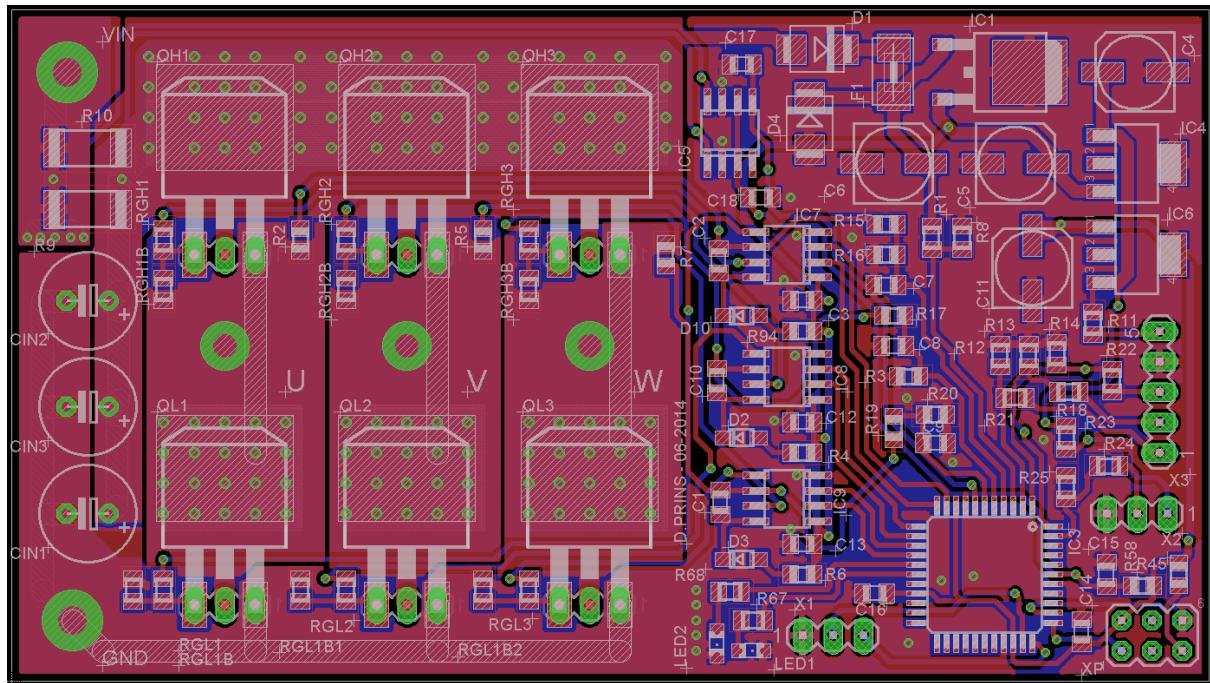
Een 12V regelaar voor de gate driver.
Een stroom meet ding + weerstanden.

De andere bevat de microcontroller, die voor elke 6 mosfets een PWM signaal produceert, die gaan door een kabeltje heen.

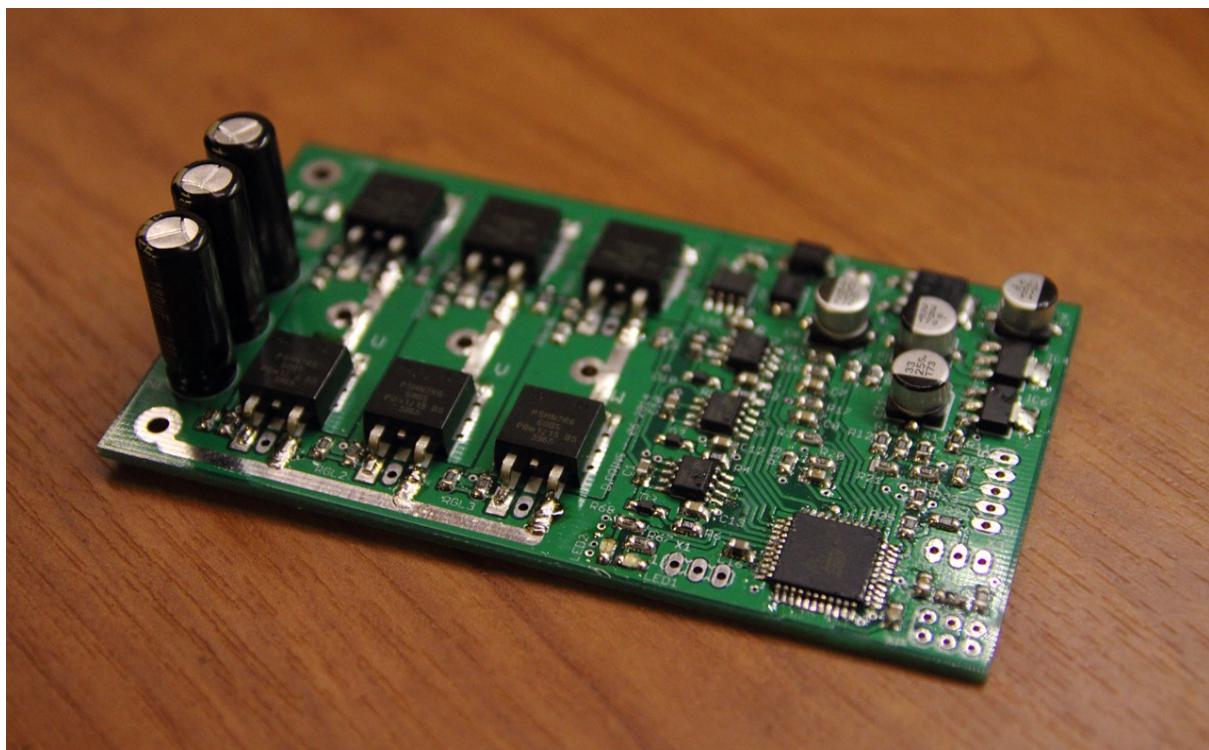
De andere kabel komt de accu spanning op binnen, gaat 5V voor vermoedelijk de stroom meting naar buiten.

Ik heb i.p.v. een enkele driver, 3 losse drivers: IRS2001.

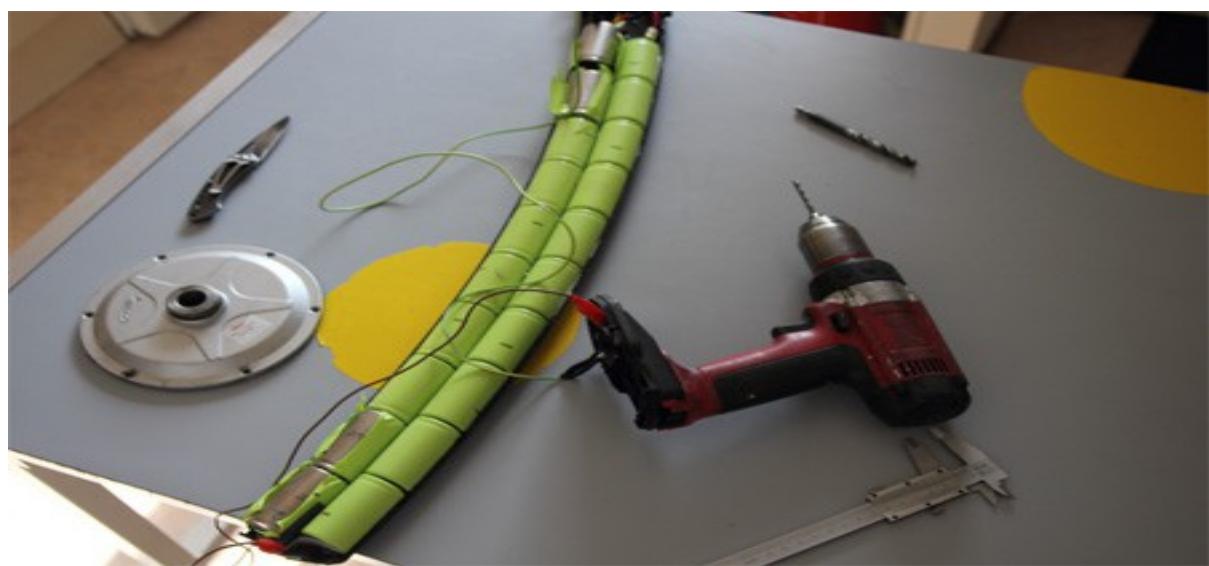
En verder heb ik alles gewoon op één printje geplaatst, zodat het makkelijker te testen is.
Vervolgens is het zo snel en goedkoop mogelijk bij een chinees besteld:



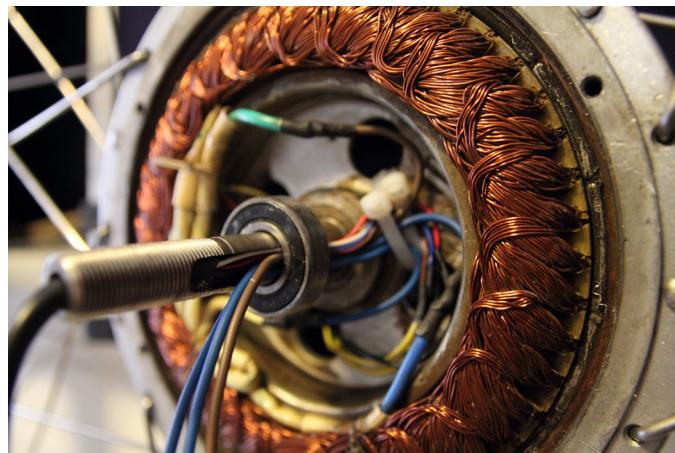
Een weekje later was PCBje binnen, en zat het in elkaar:



Om het testen makkelijk te maken, heb ik de hall sensoren (5 draadjes) plus de 3 fase draden naar buiten gehaald, zodat de motor verder dicht kan blijven, ook als ik de inbouw versie ga testen. Hiervoor heb ik een los wiel, uiteraard defect, van marktplaats getrokken. Een accu boor zonder accu uit het afval gehaald, de accu inhoud uit de de banaan aan de boor geknoopt, alles met als doel mij het gevoel te geven dat de fiets zich zelf kan kannibaliseren:



Met als resultaat:



Vervolgens is wat programmeer werk en uitzoeken welk draad in nou waar aan vast gemaakt heb, maar na een tijdje deed het in ieder geval wat:

https://www.youtube.com/watch?v=_d-LPEhaNIY

En in het volgende filmpje: <https://www.youtube.com/watch?v=T1ocqZR6tVY> heb ik de hall sensoren aangesloten, en kan de snelheid vooruit/achteruit via een console venstertje ingesteld worden.

Als mensen het interessant vinden om het schema en de code in detail besproken te hebben, laat maar weten... dan doe ik dat.

Er komen de volgende keer(en) zowiezo wat horror pics van hoe de eerste test sessie is afgelopen, hoe de nieuwe testopstelling er uit ziet, en waar dit alles uiteindelijk heen gaat.

Alles moet (kapot) draaien

Door Infant op zondag 21 september 2014 15:20 - Reacties (22)

Categorieën: Gepruts, Sparta: Fiets verbouwen tot het stuk gaat., Views: 21.874

Dit is een vervolg op [Alles moet draaien](#). Het vervolg staat hier: [Alles moet rond draaien](#).

Ik had net een stukje hardware in elkaar gefrommeld, en het jeukte enorm om het op mijn fiets te monteren.

Voordat ik dat ga doen, wou ik eigenlijk wel graag weten hoe hard ik fiets. Want... SPEED IS KEY!

Op het moment is alle orginele Batavus rommel er uit gehaald, en ligt nog ergens in een doos niks te doen.

De accu die er oorspronkelijk in zat is helemaal uit elkaar, gezien ik daar geen teken van leven meer uit kreeg. En de elektronica daarvan is tot op het bot kaal geplukt. Het leek me niet zo'n geweldige idee om die weer terug in elkaar te zetten.

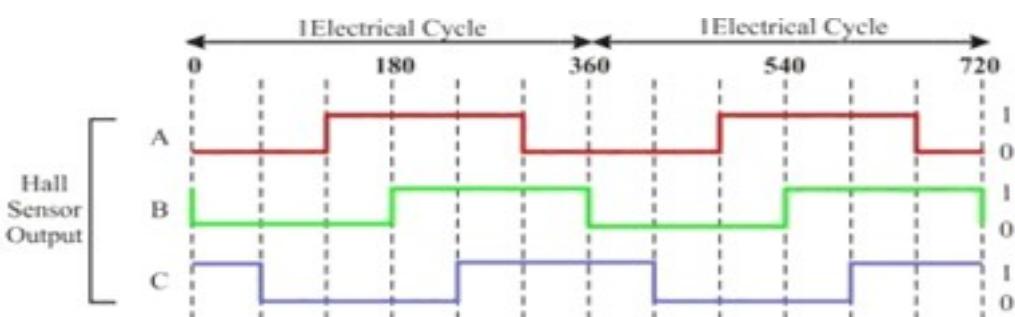
Gelukkig heb ik nog een werkende banaan, uit de debug-sessie een aantal posts terug. Als ik die kan laten werken met de orginele motor controller, kan die op zijn minst als kilometer teller ingezet worden... tenminste, dat is het plan.

Tijd om de motor controller die ik uit mijn motor gesloopt heb op te graven, en op mijn kuuroord te leggen. (Mijn bureau is verheven tot kuuroord, waar defecte elektronica tot leven gemasseerd word.)

Stap 1: Kuuroord

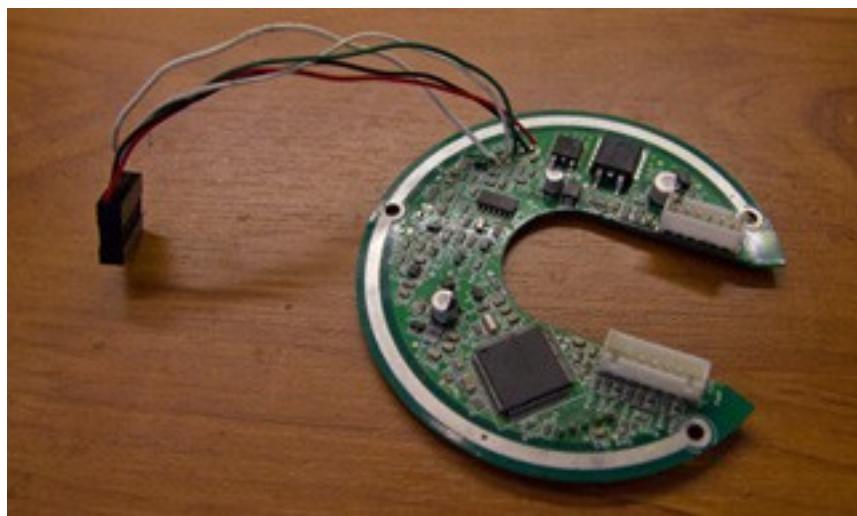
Ik ben er 99% zeker van dat de kilometer teller de hall sensoren gebruikt om vast te stellen hoe hard het wiel rond draait. Alles op dom geluk in elkaar zetten werkt bij mij nooit, dus ik pak dit stap voor stap aan.

Als je dit plaatje van de vorige keer er bij pakt...



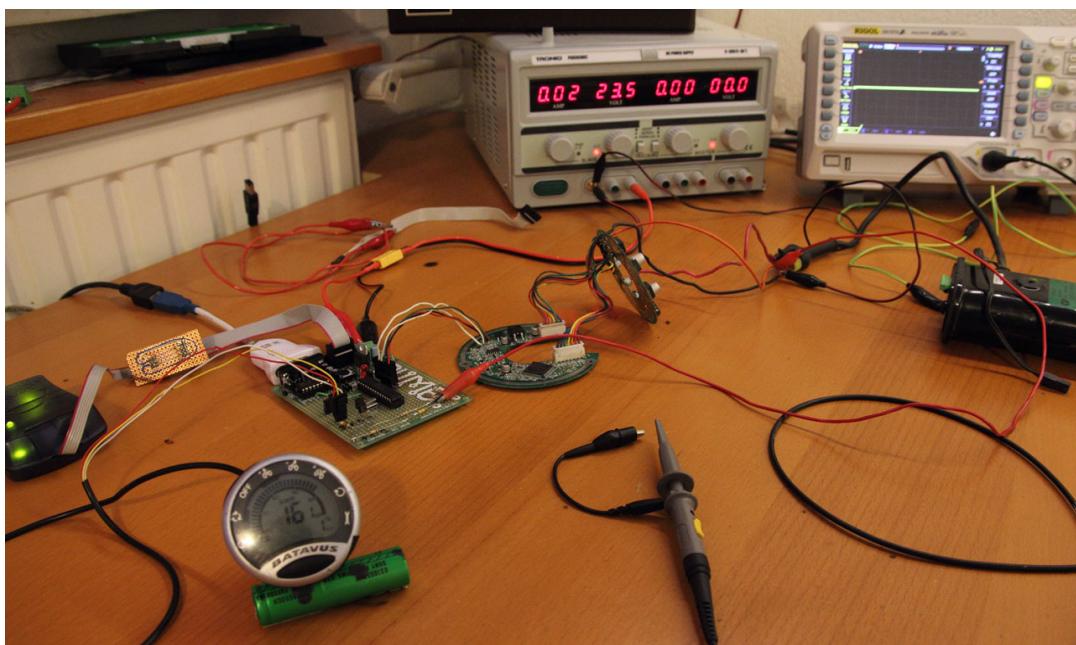
... zie je dat de hall sensoren een vrij gemakkelijk signaal produceren, makkelijk om na te doen. Dus ik dacht, als ik die gewoon zelf fabriek, en op de hall sensor ingang aansluit.... wellicht denkt hij dan dat er iets van rotatie plaats vind?

De plek waar de hall-sensoren aangesloten zaten, zijn netjes gelabeld, dus daar friemelen we een stekker aan:



Vervolgens heb ik dit (de motor controller), banaan-accu en display allemaal aan elkaar gehangen. Het Olimex bordje geeft nu continue een signaal uit zoals uit het plaatje hierboven, en luistert tussendoor al het dataverkeer af en duwt die door naar de de PC.

Met een berg draadjes zit alles aan elkaar gefriemd, en ziet het kuuroord er ongeveer zo uit:



Wat zie je allemaal? Links een programmer, het olimex bord (wat langzaam een puinhoop begint te worden) dient hier als data-snuiver. Een ourwoud van krokodillen klemmen, die de voeding en data van de accu aftappen, de twee ronde PCBjes die de zwaartekracht trotseren, een bosje van 5 draadjes die van het olimex bord naar het ronde PCB gaan. (En een firewire kabel die ook mee wil doen.)

En eigenlijk al direct na het aansluiten, kijk... onderin op het display... Kijk wat hij zegt: We gaan 16.1 km/h! Op een bureau! We hebben zelfs al 1km afgelegd! Hoe is het mogelijk?

Het kan deze motor/display combinatie kennelijk geen enkele moer uitmaken dat ik er een jaar later een andere accu op aan heb gesloten. Sterker nog: Ik heb nog nooit zo weinig foutmeldingen gezien op dit display.

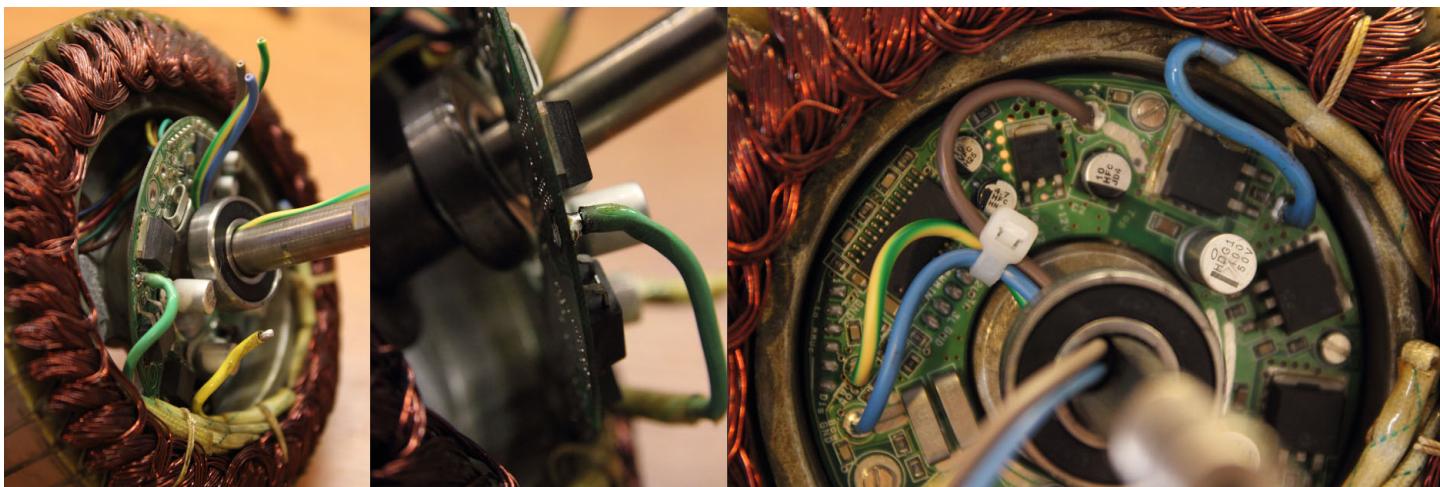
Als bijkomstigheid is het leuk om naar het dataverkeer te kijken: De accu vraagt aan het display om zijn serie nummer tegen de motor te vertellen. Kennelijk moet de motor dat dan goed vinden, en vervolgens geeft hij de snelheid en afgelegde afstand en nog wat rommel door.

De accu ziet deze data voorbij komen, en zet dit eens in de zoveel tijd op het display. Ik heb hier uiteraard een tooltje voor geschreven, want ik werd helemaal gestoord van Termite output heen en weer copy/pasten:

FROM:	B01T	TO:	MOTR	CRC AB<AB>	-	OK	-	10	02	21	09	00	AB
FROM:	B01T	TO:	40	CRC 19<19>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	40	CRC 19<19>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	40	CRC 19<19>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	80	CRC 25<25>	-	OK	-	10	84	20	25		
FROM:	B01T	TO:	80	CRC 25<25>	-	OK	-	10	84	20	25		
FROM:	B01T	TO:	80	CRC 25<25>	-	OK	-	10	84	20	25		
FROM:	B01T	TO:	MOTR	CRC AB<AB>	-	OK	-	10	02	21	09	00	AB
FROM:	B01T	TO:	00	CRC EEEEEE	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	00	CRC EEEEEE	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	00	CRC EEEEEE	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	00	CRC EEEEEE	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	00	CRC EEEEEE	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	00	CRC CC<CC>	-	OK	-	10	04	20	CC		
FROM:	B01T	TO:	MOTR	CRC CC<CC>	-	OK	-	10	02	21	09	00	AB
FROM:	B01T	TO:	40	CRC AB<AB>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	40	CRC 19<19>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	40	CRC 19<19>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	80	CRC 25<25>	-	OK	-	10	84	20	25		
FROM:	B01T	TO:	80	CRC 25<25>	-	OK	-	10	84	20	25		
FROM:	B01T	TO:	80	CRC 25<25>	-	OK	-	10	84	20	25		
FROM:	B01T	TO:	MOTR	CRC AB<AB>	-	OK	-	10	02	21	09	00	AB
FROM:	B01T	TO:	MOTR	CRC 75<75>	-	OK	-	10	01	20	32	75	
FROM:	DISP	TO:	MOTR	CRC A4<A4>	-	OK	-	10	02	20	09	10	00
FROM:	DISP	TO:	MOTR	CRC C8<C8>	-	OK	-	10	02	20	09	10	00
FROM:	DISP	TO:	MOTR	CRC 2F<2F>	-	OK	-	10	01	21	34	01	7F
FROM:	B01T	TO:	A0	CRC EEEEEE	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	00	CRC EEEEEE	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	00	CRC EEEEEE	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	00	CRC EEEEEE	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	MOTR	CRC CC<CC>	-	OK	-	10	04	20	CC		
FROM:	B01T	TO:	MOTR	CRC AB<AB>	-	OK	-	10	02	21	09	00	AB
FROM:	B01T	TO:	40	CRC 19<19>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	40	CRC 19<19>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	40	CRC 19<19>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	40	CRC 19<19>	-	OK	-	10	44	20	19		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	60	CRC D2<D2>	-	OK	-	10	64	20	D2		
FROM:	B01T	TO:	80	CRC 25<25>	-	OK	-	10	84	20	25		
FROM:	B01T	TO:	80	CRC 25<25>	-	OK	-	10	84	20	25		
FROM:	B01T	TO:	80	CRC 25<25>	-	OK	-	10	84	20	25		
FROM:	B01T	TO:	MOTR	CRC CC<CC>	-	OK	-	10	04	20	EE		
FROM:	B01T	TO:	MOTR	CRC AB<AB>	-	OK	-	10	02	21	09	00	AB

Al mijn tootjes doen meestal één ding, en over drie maanden weet ik niet meer wat dat was. Het lijkt erop dat de accu met nog een aantal dingen probeert te babbelen, die er niet zijn.

Dat is allemaal heel leuk en enig, maar nu eerst: Inbouwen die handel. Wie weet, doet hij het straks gewoon weer?



Op het tweede plaatje vraag je je af: Zit die dropveter niet wat karig? Nee, want er moet ook nog een isolatie flapje tussen. En dat werkt niet als je het draadje er netjes doorheen doet.

Nu begin ik een beetje te begrijpen waarom deze dingen zo duur zijn. De fiets zelf zit het hem niet in: Dit type motor is zo'n beetje het goedkoopste wat je kunt maken. Het heeft verder geen versnellingen, en de verlichting is een standaard 6V gloei lampje. De spatborden laten los, en het plastic om de kettingkast ratelt als een zak grind in een wasdroger. Zo'n beetje elke schroefje en boutje heeft een andere maat, en ze zijn allemaal verroest.

Op zich is het verder een prima ding, ik rijd er graag mee trappetjes op en af, en hij doet het nog steeds.

Maar... het gros van de kosten zal wel zitten in het in elkaar zetten van dit ding. Wat een pokke werk. Alles past precies net niet, en moet meet een zekere hoeveelheid geduld terug in elkaar gedwongen worden.

Nu is het printje terug gesoldeerd, de motor weer dicht en achterin de fiets. De "nieuwe" accu zit terug in het frame, en het display zit weer op zijn plek. Eens kijken wat hij doet....

Ik rijd naar buiten, zet de ondersteuning op standje 3, (klik hoor je dan) en begin te trappen. Meteen geeft hij keurig de snelheid aan. Maar ondersteunen doet hij nog niet echt.

Maar...

Ik zal wel iets verkeerd om aangesloten hebben, want op het moment dat ik ophoud met trappen gaat hij er uit zichzelf vandoor, en het moment dat ik weer trap stopt hij met mee helpen. 😊

...

Dat is vrij irritant als je een helling op fietst. Het moment dat je ene voet kracht zet, stop hij, dan heb je even zo'n punt dat beide voeten niks doen, waarin hij eventjes mee helpt, en dan bij de volgende voet doet hij weer niks.

Nou heb ik ook op een berg werkende exemplaren van deze fiets gereden, en ik vind trap ondersteuning sowieso irritant: Het moment dat je ophoudt met trappen gaat hij nog even eigenwijs één seconde door, en als je weg rijdt moet je lekker even wachten.

Maar dit boeit me allemaal niet. Zonder trapondersteuning doet de snelheid en afstandsmeter het ook, en dat was het doel. Next!

Stap 2: Meer mechanisch geneuzel.

Stap twee is de MUT (Motor Under Test) aansluiten. Waar gaat die? Juist, in het voorwiel. Waar anders?

Meteen loop ik weer tegen een probleem op: de as diameter achter is anders dan voor. Achter is 12mm, en het voorwiel is 10mm. (Waarom kan dat niet hetzelfde zijn? Nu heb ik twee types moertjes en sleutels nodig.)

Gelukkig is het frame en de voorvork van aluminium gemaakt, en dan ook nog aluminium van het type zacht als boter. (Ik weet niet welk DIN nummertje dat is...) En is echt twee seconden werk voor de banaan-powered-power-tool (BPPT). (Door een boor van 14.4V op 24V te draaien gaat alles sowieso veel sneller, en houd hij je hand ook lekker warm.)

Probleem twee is dat wiel wat ik voorin wil hebben, gewoon echt veel te vadsig is en er niet tussen past. Het voornaamste wat er dan in de weg zit is het tandwiel, en een stuk aluminium aan de andere kant... waar allerlei accessoires zoals remtrommels op geschroefd kunnen worden.

Ik heb niet zoveel tools. Een veil, een metaal zaag een berg schroevendraaiers en ongeveer 10 boren zonder accu... en een bezem. Die gaan dit allemaal niet zo makkelijk kunnen doen.

Maar deze fiets heeft een ongekend vermogen om zichzelf op eigen kracht kapot te

maken: Door de motor terug in de opstelling te plaatsen, en de zelfbouw-controller te vertellen dat ik continue een even grote hoeveelheid stroom de motor in geperst wil hebben, krijg je een constant koppel. En dat werkt ideaal als draaibank. Zo hoef ik de metaal zaag er alleen maar op te houden, en dan vreet het overtollige aluminium zichzelf langzaam op:



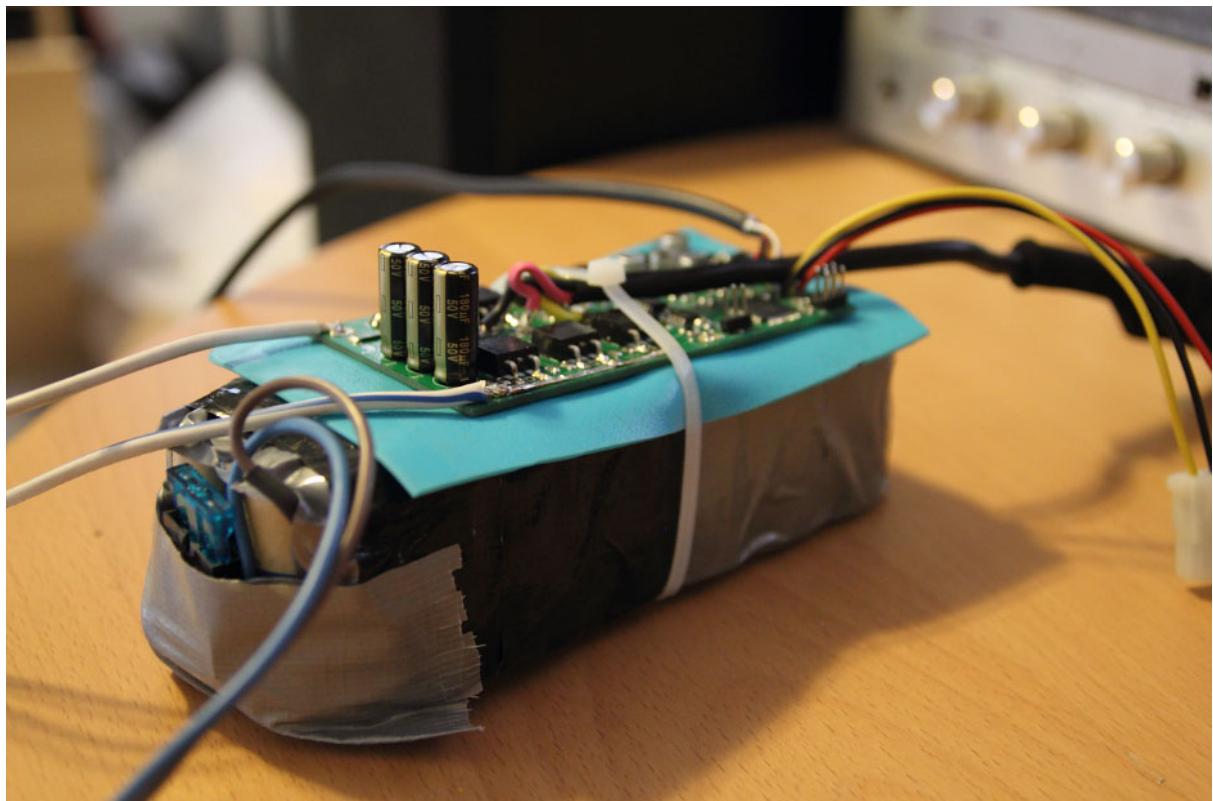
Na een minuutje of 10 waren beide stukjes aluminium er af, en met een overmatige hoeveelheid spacers paste het eindelijk in de voorvork.

Volgende probleem: Ik was iets te enthousiast met aandraaien, en nu sprong het ding wat de motor op zijn plek moet houden (een torque washer... is in het nederlands?) van ellende uit elkaar.

Pfff. Ik wil gewoon testen. Goed vast draaien, en hopen dat het blijft zitten.

Laatste stap: Deze motor + controller moet natuurlijk ook prik hebben, en het zou zonde zijn om de gloednieuwe banaan in het frame open te maken. (Ze zouden eigenlijk van kleur moeten veranderen, groen - geel - bruin naarmate je ze leeg trekt.)

Nou heb ik nog een ongebruikte 24V NiCd accu uit een boor in de kast liggen, die verder nergens op past. Die kan opzich wel dienst doen. Dit leverde na niet al te lange tijd de volgende contraptie op:



Dat blauwe spul is een silicone matje wat extreem precies op maat geknipt is (Not). Links zit een zekering zodat ik 's nachts ietsje beter slaap, en verder heeft het alle standaard items die je moet willen hebben: duc(k)(t)ape en tie-wraps voor gegarandeerd mechanisch succes.

Dit dient vervolgens met dezelfde hoogwaardige mechanische tools tegen het frame gemonteerd te worden, met als eind resultaat:



Werkelijk prachtig. What could possibly go wrong?

Het controller printje was voorzien van de meest basis functionaliteit: het gas hendeltje uitlezen. (Thumb throttle van een chinees op eBay, mag je lekker 5 weken op wachten.) Vervolgens werd dit vertaald naar een combinatie van snelheid en stroom: 50% hendel betekend 50% van de maximale stroom en 50% van de snelheid. (Ongeveer...)

Dit ging best wel aardig.. je voelde dat je vooruit getrokken werd. Maar de 5 Ampère waar de stroom limiet op ingesteld stond leek op de test opstelling gigantisch veel, (het heeft al vrij veel aluminium krom gebogen) maar als je er vervolgens met je kont bovenop gaat zitten stelt het niet zo veel meer voor.

Dus dat was na 2 minuten rondjes rijden al niet interessant meer, dus dan maar eventjes om programmeren en gewoon de stroom limiet op veel: 20 Ampère.

Nou is 24V en 20 Ampère op zich niet zo veel, maar in stilstand vertaald dit zich naar een veelvoud daarvan aan stroom richting de motor...en dat moet allemaal door die arme draadjes van 1.5mm² heen. En die worden daar een beetje warm van.

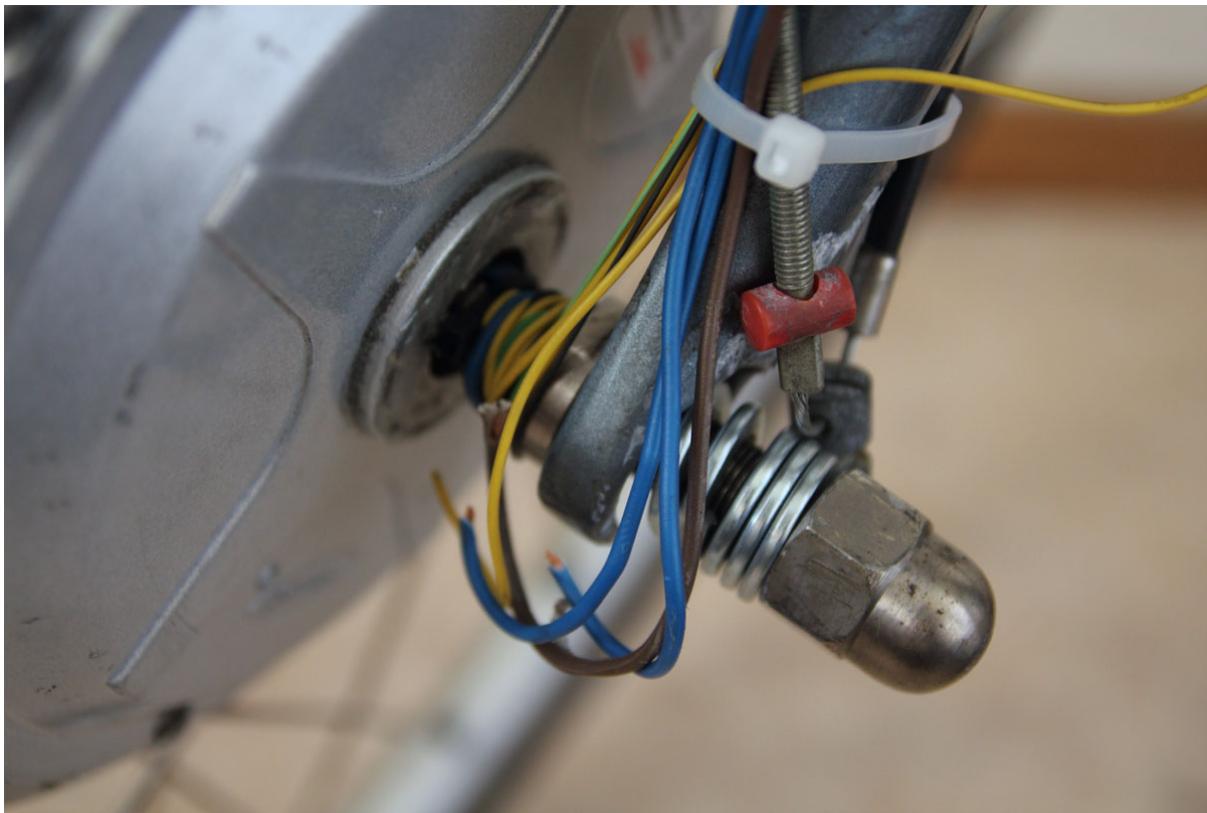
Het reed wel een stuk beter, uit zich zelf haalt hij zo ongeveer 25 km/h. Het duurt even voordat je daar bent, en bij elke vorm van wind of helling zakt het in naar 20 km/h.

Als ik dan ook de achter motor aan zet, gaat het opeens als een raket, en haal je tegen de

30 km/h. De motor achter ondersteunt tot 27 km/h, en houd dan op, de voor motor kan er dan nog net 3 km/h bij krijgen.

Toen dacht ik, kom... laat ik ook eens mee trappen. Dit heeft bij 30km/h geen zin op dit ding, want je benen moeten dan modus roadrunner mee draaien.

Zo heb ik 15 km gefietst, als ik het display mag geloven, en toen viel het me op dat de draadjes naar de motor toe steeds korter werden. Vrij snel daarop volgend werd de rit iets minder comfortabel:



Alle draden uit de motor waren kapot, en de uiteindjes zaten tegen elkaar aan kortsluiting te maken. (Vandaar het schokkerige gevoel.)

De draadjes uit de controller zaten ook tegen elkaar, en de hall sensor draadjes waren ook niet echt jofel meer.

Ik vond dat hij het nog lang uit heeft gehouden. Het goede nieuws is dat de controller deze kleine mechanische tegenslag heeft overleefd.

Zodra hij geen hall sensoren meer ziet (Omdat bijv. iemand de draadjes om een as gewikkeld heeft.), stopt hij met stroom de motor in te sturen. En al waren de hall-draadjes nog heel, dan zou hij rustig de draden naar de motor op gewarmd hebben omdat er bij kortsluiting veel te veel stroom zou gaan lopen.

Conclusie/bevindingen

- Je moet alles vast willen maken. Een torque arm of torque washer is een must, deze motoren vreten aluminium als ontbijt.
- De motor krijgt het vrij warm als je er 500 Watt in duwt.
- Een motor in het voorwiel rijdt voor geen ene meter. Want ook als ik hem als normale fiets gebruik, gaat dit ding helemaal zijn eigen leven leiden als ik het stuur los laat. Elke bocht voelt het alsof hij recht door wilt.
- Twee motoren gaan harder dan 1 motor. (Duh.)
- Gashendels zijn awesome, maar ik zet hem eigenlijk de hele tijd op 100%. Dus het had net zo goed een knopje kunnen zijn.
- Trapondersteuning is suf.
- Je moet twee remmen willen hebben. (De voor-rem zit niet vast, gezien dat een trommelrem is.)
- Trommel remmen zijn poep.
- Deze methode van testen is sub-optimaal en kost veel te veel tijd. De rij/pruts verhouding is niet goed. (4 uur prutsen, 20 minuten rijden).
- Ik moet meer tie-wraps kopen.

Op naar het volgende deel.

Alles moet rond draaien.

Door Infant op maandag 2 maart 2015 18:11 - Reacties (21)

Categorieën: Gemod / Gefix, Sparta: Fiets verbouwen tot het stuk gaat., Views: 12.767

Dit is een vervolg op [Alles moet \(kapot\) draaien](#), een eeuwig voortdurende serie over gepruts aan elektrische fietsen.

De hele serie in chronologische volgorde:

[1\) Dooie Fiets](#)

[2\) Banana?](#)

[3\) Bananen Taal](#)

[4\) Cyclic Banana Checks](#)

[5\) Alles moet draaien](#)

[6\) Alles moet \(kapot\) draaien.](#)

Je zit nu hier: [7\) Alles moet rond draaien.](#)

De vorige post hield op met een voorwiel-motor die zichzelf helaas heeft opgegeten. Omdat ik toch ergens heen moest fietsen, heb ik het onbemotorde (*dit is nu een*

woord) voorwiel maar weer terug gezet.

Eigenlijk moet ik twee fietsen hebben: Eentje die in permanente werkende staat verkeerd, en eentje die als rollend test exemplaar dienst kan doen om uit te vinden bij welke volgende stap er (onvermijdelijk) weer iets stuk gaat.

In eerste instantie zocht ik op marktplaats een beetje rond naar nog een overleden accu, want daar kunt je er nooit genoeg van hebben. Op een goede dag kwam er eentje voorbij, deze zat zelfs nog in originele verpakking (d.w.z: met fiets eromheen) en kostte maar 50 Euro.

"I need it. Buy it. Store it."

Het ging hier om een mannetjes fiets. Het frame voelt daar een stuk minder wankel van, en een ander mechanisch voordeel is o.a. dat de achtersas twee vlakke stukjes heeft. En dat betekent dat de motor extra vast kan.

Verder heeft deze blok remmen, zodat het achterwiel een stuk gemakkelijker los te maken is. Kortom, prachtige rijdende testomgeving. Het enige wat het mist is een draadloze internet verbinding.

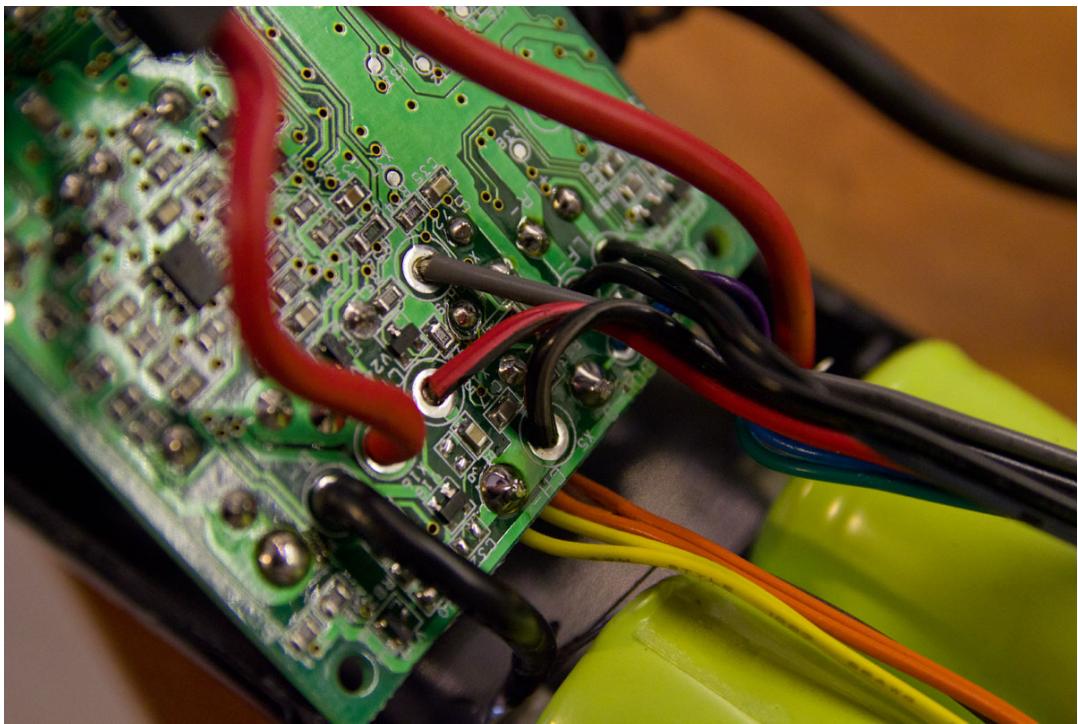
Volgens de verkoper deed de fiets het op zich nog wel, maar wou de accu niet opladen. Dat is eigenlijk niet kapot genoeg, want als het eenvoudig te maken is... moet ik het weer verkopen. Dat zou zonde zijn.

Bij het aansluiten van een lader gebeurde er eigenlijk niks. Om hem dan weer een beetje tot leven te wekken, moet je een zwengel aan het wiel geven. Hierna gaat het display aan om te zeggen d.m.v. E0001 dat de accu leeg is.

Ja. Dat weet ik. Daarom zit er een lader in, okee?

Misschien heeft hij gewoon wat "quality alone time" nodig, dus met lader heb ik hem maar even alleen gelaten. Maar een half uurtje later, was er nog niet echt iets gebeurd....

Dit gaat niet werken. Open maken die handel:



Deze ziet er verrassend netjes van binnen uit. Er zit zelfs geen coating om de print heen, waar de vorige nog wel iets van smurrie hadden om water buiten te houden. Maar ach, het is niet alsof dat iets doet.

Het eerste wat ik door mat (ja, meten moet kennelijk naar dit vervoegd worden. Ik heb het ook niet verzonnen.) is dat de 34V van de lader, niet echt op de daarvoor bestemde plek op het printje komt.

Ergens tussen de aansluiting, en het printje gaat iets fout. Er zit een draad tussen met een bobbel er in, met daarin een overleden zekering. Dat is makkelijk opgelost.

Toen dit gefikst was (lees: vervangen door een draad), ging het display aan en kwam met de melding E0022 op de proppen, wat zoveel betekent als "Laden start niet".

Hm. Eigenwijs stuk rommel.

Als last resort heb ik de lader (wat gewoon mijn brakke labvoeding is) maar op de min van de print aangesloten, en de plus direct op de accu. De stroom die dan de accu in loopt, loopt ook door de meetweerstand op dit bordje heen. Dit is niet echt iets wat normaliter gebeurt, maar het zou zomaar kunnen dat hij die capaciteit gaan zitten meten en daar iets mee doet??

Even later was hij direct van mening dat de accu vol was:



Helaas was hij ook zo de kluts kwijt, dat er met geen mogelijkheid iets anders als bovenstaan plaatje op het display te krijgen was. Ik had op zijn minst verwacht dat hij nu over het display serie nummer ging zeuren.. maar ik krijg er met geen mogelijkheid iets zinnigs uit.

Okee, ik verklaar het een succes! Ik heb het gesloopt. Het is niet meer voor de verkoop geschikt, dan mag het nu helemaal uit elkaar. 😊

Dus net als de vorige keer, heb ik ook hier het zekeringetje netjes verbonden en dezelfde dropveter aan de banaan gemaakt, zodat het motor control printje makkelijk los kan, en de accu makkelijk opgeladen kan worden.



Volgende stap is de motor uit elkaar halen, dit heb ik nu zo vaak gedaan dat ik er een speciale tool voor gekocht heb:



Normaliter houd ik niet van tools die maar één ding kunnen. Bijvoorbeeld een ei-in-plakjes-snijder, een boon-in-stukjes-maler, een vlapak-leeg-schraper of een soeplepel met duizend gaten er in. Allemaal kansloze voorwerpen die door een mes en/of lepel vervangen kunnen worden. Ik kan er niet tegen.

Behalve dus dit voorwerp. Het is een Poelietrekker, kennelijk. Dit levert per operatie een tijdwinst van c.a. 1 uur op. Werkelijk geweldig. Het is verder helaas verder niet voor iets anders in te zetten.

Deze motor ziet er werkelijk het beste uit van alle die ik tot nu toe open gemaakt heb (hij draaide ook gewoon nog, wat al een goed teken was). De magneten zijn nog niet vermalen tot stof, en de elektronica ziet er dus ook nog netjes uit.



Dus... zelfde verhaal: printjes er uit, en de hallsensoren + 3 fasen draden door de as naar buiten moffelen. De stekker waar de motor mee vast zit is een AMP Superseal (Turbo zeehond), en die zijn los op fanell te koop. Voor de Hall sensoren heb ik daar een mooi 5 polig stekkertje gevonden, zodat het allemaal weer makkelijk los kan:



Zo, nu we dat gehad hebben, kan ik de laatste dingen aan het display doen:

Meer met het display communiceren.

Origineel kun je met het display op deze fiets bar weinig instellen. Het geeft de snelheid aan, je kunt 3 standen instellen, en er wordt ook nog iets van afgelegde afstand bijgehouden. Als klap op de vuurpijl kun je met het knopje bovenin de verlichting aan een uit zetten.

Als je tijdens het rijden het display er vanaf trekt, stopt hij na een aantal seconden met ondersteunen. Het werkt dus als een soort slot. Dat wil ik ook.

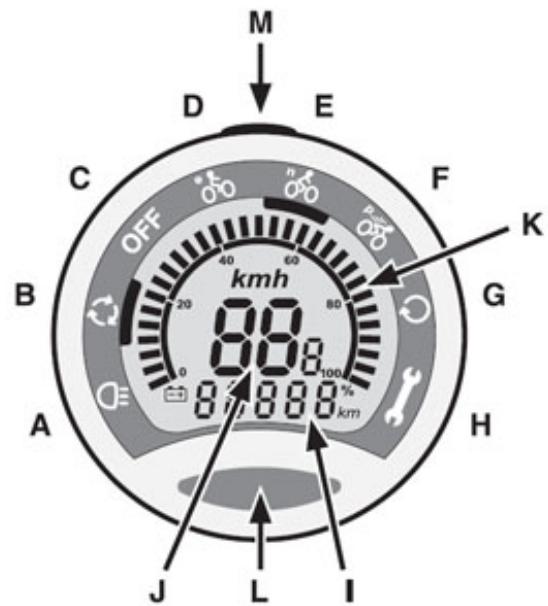
Als je er een display met een ander serie nummer op prikt, vertikt hij het om te gaan werken. Dat soort onzin heb ik natuurlijk allemaal niet nodig.

Bij elke van de 3 standen, staat een segmentje om aan te geven dat je in die stand zit:

2.2. Knoppen en indicaties

Iedere geselecteerde functie wordt aangegeven door een balkje.

- A Verlichting
- B Dagafstand
- C Trapbekrachtiging OFF
- D Trapbekrachtiging ECO
- E Trapbekrachtiging NORMAL
- F Trapbekrachtiging POWER
- G Totale afgelegde afstand
- H Onderhoudsindicatie
- I Variabele indicatie
- J Snelheid
- K Vullingsgraad batterijpakket
- L Setknop
- M Lichtknop



Dit is mooi, maar met 4 segmenten (C tm F) kun je natuurlijk veel meer. Want 2^4 is 16, dus er kunnen wat mij betreft gewoon 16 standen op... waar stand 0 natuurlijk uit is.

Verder geeft het getal in het midden van het display ten alle tijde de snelheid aan, dat wil ik op zich ook wel.

Mijn ervaring is dat 95% van de tijd de onderste regel een van de vele error codes aangeeft, maar mijn ervaring is wellicht iets gebiased. De overige 5% van de tijd geeft het de afgelegde afstand aan.

Dat vind ik op zich niet zo heel boeiend, als ik wil weten hoe ver ik rijd heb ik daar GPS voor. Ik wil eigenlijk wel ten alle tijden zien hoeveel stroom er loopt, en de snelheid zou eigenlijk wel de accu spanning kunnen weergeven.

De knopjes uitlezen had ik nog niet echt gedaan, maar het is (achteraf) vrij eenvoudig. Ik quote uit een aantal posts terug:

Het een na laatste getal loopt steeds op:

10 C1 21 22 01 6F

... 02 9F

03 0E

04 3C

...

En na 0F begint hij weer opnieuw.

De accu stuurt dit bericht naar het display toe, en wordt altijd beantwoord in de vorm van 10 22 C2 22 00 XX XX, behalve als je een knop ingedrukt houdt. Dan veranderd de 0 in een 1 als de bovenste knop wordt ingedrukt, een twee als de voorste ingedrukt wordt, en 3 als je beide ingedrukt houdt.

Als ik steeds hetzelfde bericht stuurt, gebeurt er niks, maar als ik ze laat oplopen tot 15 werkt het prima. Geweldig.

Hier een impromptu filmpje, waar ik de knop op het display tijdelijk als throttle input gebruik. (Filmen met één hand, zonder autofocus is moeilijk. Alvast excuses.) Met het bovenste knopje kan ik nu door menu's heen wandelen, en met de onderste kunnen daar oneindig veel arbitraire waarden aan arbitraire variabelen geknoopt worden. (Als je uit het menu gaat, slaat hij ze op in het rom gedeelte van de micro controller.)

In de huidige toestand doet het display het volgende:

Je kunt door een menu heen bladeren door de bovenste knop in te drukken. Als je die 2

seonden met rust laat, geeft hij weer de snelheid aan.

Menu 1 kun je instellen of de snelheid of accuspanning weergegeven moet worden.

Menu 2 stel je de remkracht mee in.

Menu 3 een of andere test variabele...die elke dag anders is.

Menu 4 is het mooist: Als je daar 4 keer op druk, gaat icoontje G aan.

Die heb ik het "road-legal == false" icoon genoemd (RLIF Icoon). Alle features die op deze fiets niet aan de wet voldoen (dat zijn zeg maar alle features) worden geactiveerd als dat icoontje aan staat.

Het moment dat je het display los trekt, of weer 4x in dat menu op het knopje drukt, gaat hij weer naar road-legal toe... en wordt het weer een doodsaai fiets waar je zelf al het voortstuwwerk moet doen.

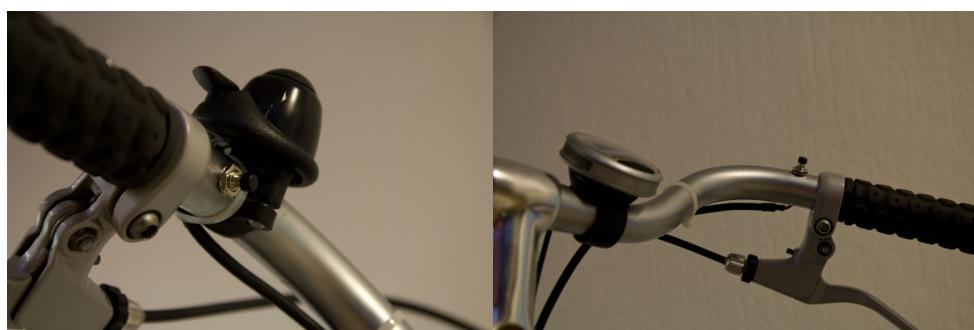
Dit leek me een extreem handige feature, mocht ik op straat een flits-snor tegen komen.

Die kan ik dan vol overtuiging vertellen dat alle ductape, tiewraps en los bengelende elektronica geen enkele functie heeft, en puur als decoratie dient.

Verder geeft de kilometerstand onderin nu de stroom aan, als je de rem knop indrukt gaat hij regeneratief remmen, en wordt de stroom negatief. (Er zat zelfs een - karakter in het display, aardig he?)

De accu volheids- balk gebruik ik om de hoeveelheid throttle aan te geven.

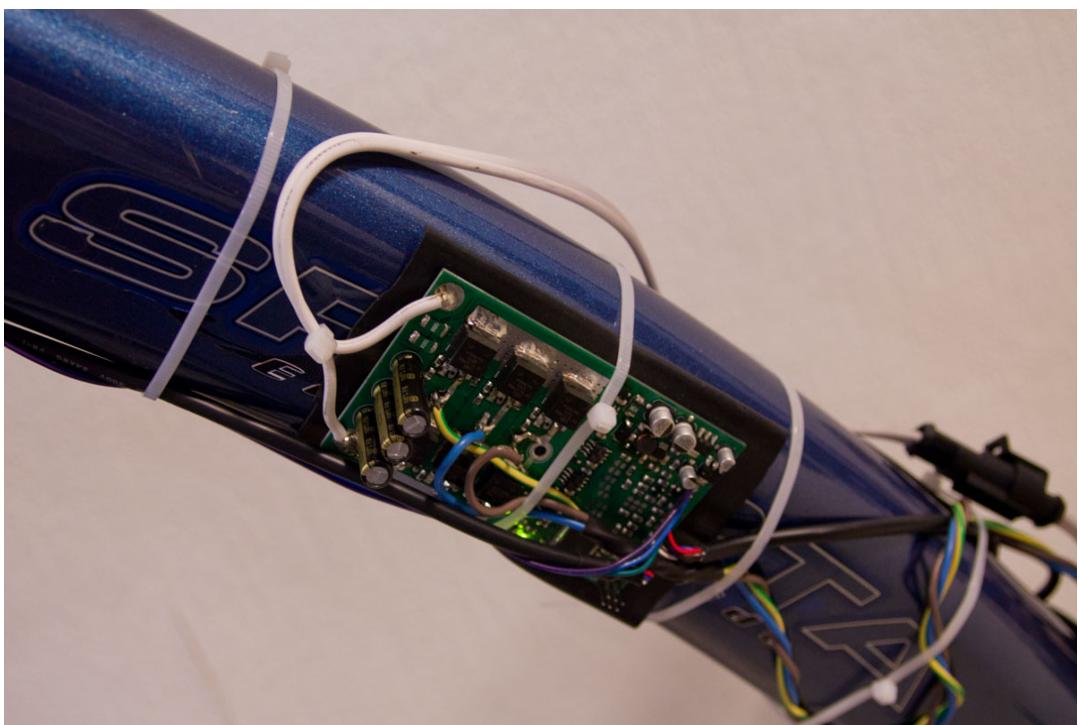
De gas hendel heb ik vervangen door een knopje, want ik kwam er al vrij snel achter dat minder dan 100% gas geven niet vaak voor komt. Nu hoef ik niet 6 weken op Chinese een gas hendel te wachten als ik hem sloop. Het remmen gaat met eenzelfde knopje, en ze zitten nu zo in het stuur weg gewerkt:



Links zit het rem knopje, rechts het gas knopje. Als je het stuur normaal beet hebt, hoef je alleen je duim in te drukken, en er gebeurt wat. Het 100% gas signaal word afhankelijk van de stand vertaald naar een veelvoud van 4 Ampère motor stroom: Waar stand 6 er maximaal 24 Ampère naar toe stuurt.

Wat ook handig is, is de cruise control: Als je 4 seconden het gas vol ingedrukt houd blijft hij op 100% staan, en kun je het knopje weer los laten. (Want... anders wordt je rechter duim moe.). Als je dan remt (of als je de rem knop gesloopt hebt: het display los haalt) schakelt hij de cruisecontrol uit.

Wat mij betreft zitten alle features er nu in. Het test printje heb ik secuur tegen het frame gemonteerd:



Ik heb hier, met wat extra ductape eromheen uiteraard, in de tussentijd al honderden kilometers mee afgelegd, en ik moet zeggen dat het goed werkt. (Je kunt er gewoon mee door de regen en hagel fietsen. Ductape is geweldig.)

Maar toen... werd het winter, en akelig koud buiten. Bij het weg rijden ging het display na 3 minuten uit zichzelf van "road-legal == false" stand af. Het scherm updateerde niet meer, en het reageren op de knopjes ging moeizaam en traag.

Issie nu kapot?

Valt mee. Het communiceren met het display gebeurt via een seriele interface. Deze communiceren op een vaste snelheid, 9600 bps. De klokfrequentie die de microcontroller nodig heeft om 9k6 te halen, is meestal niet helemaal een mooi veelvoud van de basis klok frequentie (in dit geval 32Mhz). In de code die er op draait, stel je een deler in om van die 32MHz op de juiste frequentie uit te komen.

Nou is dat niet zo'n probleem, maar omdat ik geen zin had om een extern kristal te gebruiken, liep de microcontroller van de interne klok, en die heeft ongeveer de volgende temperatuur afhankelijkheid:



Links de klokfrequentie in Hz voordat het de PLL in gaat, rechts de temperatuur in graden fahrenheit (x2), onderin de stand van de maan in eenheden van $[2\pi/1 \text{ dozijn}]$. Zoals je ziet: een grote puinhoop.

De oplossing is echter vrij simpel: Omdat het display telkens antwoordt geeft als hij correct een bericht heeft ontvangen, kan de afwezigheid van antwoord drie dingen betekenen:

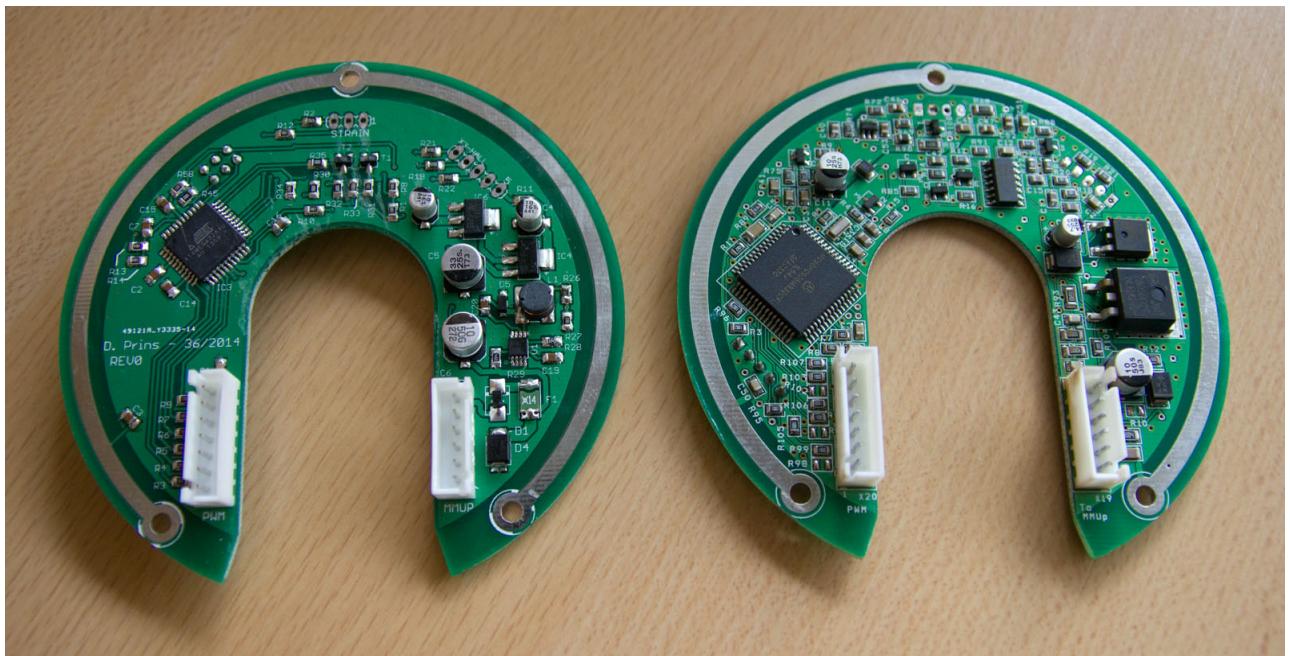
- 1: Het display is stuk.
- 2: Het display is weg.
- 3: Het vriest (beter gezegd: het is niet kamertemperatuur.)

In alle drie de gevallen, past de motor controller de communicatie frequentie naar het display steeds een fractie aan, net zolang tot er weer antwoord komt.

En nu?

Ik heb een wat nettere oplossing gemaakt. Van alle losse motoren die ik ondertussen heb, deed het power gedeelte het nog prima. Dus ben ik als een Chinees aan de slag gegaan, en heb het logica gedeelte van mijn controller wat ronder gemaakt, zodat het nu plug-and-

play in een motor kan. Ik hoef alleen de dop er af te halen, het printje los te maken, en die van mij er in te ploppen:



Drop in replacement.

De gas en rem inputs gaan via een extra commando, over het communicatie draad naar de motor toe. De trapondersteuning optie zit er wel op, maar die gebruik ik nooit.

Op het moment dien ik zelf als batterij management systeem. Bij 18V stopt hij uit zichzelf met de accu leeg zuigen. Door de accu gewoon helemaal los te koppelen kan je hem ook

maanden in de schuur laten staan zonder dat hij telkens leeg loopt, of 24/7 aan de peut moet hangen.

De accu is op zich nog vrij aardig, er zit van de 10Ah nog iets van 8Ah in. Hij wordt lekker warm als je er 500W uit haalt.

Een grotere accu zou op zich geen kwaad kunnen.....

Alles moet rond draaien - Deel 2

Door Infant op maandag 13 april 2015 17:20 - Reacties (16)

Categorieën: Gemod / Gefix, Sparta: Fiets verbouwen tot het stuk gaat., Views: 7.324

De titel suggereerde het al een beetje: dit is een vervolg op het vorige deel:[Alles moet rond draaien.](#)

Sinds de laatste blogpost, heb ik een bovengemiddeld hoge (Lees: meer dan 0) hoeveelheid mailtjes en berichtjes gehad, van mensen die opgescheept zitten met eenzelfde wrak als waar ik mee begon, met vragen zoals: "Hoe?", "Wat?", "[Waarom?](#)" en "[OMG Help!](#)"

Het beeld wat ik had van de marktpenetratie van deze fietsen wat voorheen alleen maar gebaseerd op de 55+ er's die ik elke dag voorbij rijd, waarvan het denk ik niet overdreven is te stellen dat ~50% een elektrische fiets heeft. Dit beeld is nu wat bijgesteld.

Wat ik dagelijks zie rond rijden zijn namelijk overduidelijk alleen maar de werkende exemplaren...

Marktplaats bevat op het moment van schrijven 25 pagina's aan nog rijdende Sparta Ion's. (Waarvan een grote kans dat de helft eigenlijk stiekem niet werkt.) Verder is er nog maar 1 pagina met defecte omdat ik als een woeste aap steeds alle reserve onderdelen er vanaf sloopt.

Het begint me nu echter duidelijk te worden dat heel Nederland verzadigd is met fietsen. Ons land puilt er mee uit. Werkelijk elk dorp bevat **Silos** vol met Batavus en Sparta ION wrakken. Van 50% is het BMS (Battery Murdering System) ervan overtuigd dat hij permanent leeg is, van de andere 50% is de elektronica beschimmeld, de motor vastgeroest of een klein ongedocumenteerde component van 1 cent is spontaan opgerookt.

Bizar.

Iedereen die mijn tot nu toe benaderd heeft, en met zo'n lijk opgescheept zit wil er kennelijk, net als ik, zo min mogelijk tot 0 Euro aan uitgeven. Vervolgens weekenden prutsen met als doel dat het dan weer rijd. Niemand wil dat op dezelfde manier. Het moet natuurlijk allemaal net iets anders. De overlap die ik uit alle emails tot nu toe heb kunnen vinden:

- Het moet trapondersteuning doen.

Deze mensen hebben duidelijk niet grondig genoeg het voorgaande oerwoud aan tekst gelezen, wat ik overigens niemand kwalijk neem. Ik weet zelf niet eens meer waar het over gaat. Maar een CLTR+F en "trapondersteuning" brengt me naar iets wat ik een tijdje terug kennelijk geschreven heb:

...bla bla bla ... hel en ellende....

Conclusie/bevindingen

- Trapondersteuning is suf.

Ik had op de hardware uit de vorige post wel ruimte over gelaten om de trapas sensor op aan te sluiten, maar had er verder niet echt naar gekeken. Misschien dat ik er ooit nog eens wat mee doe. Nou, bij deze.

Hoe zou het moeten werken?

Er moet kracht gemeten worden, trapkracht om precies te zijn. Dit kan je op meerdere manieren doen, en simpelweg omschrijven dat het op meerdere manieren kan, zoals ik nu dus doe, kun je met nog wat extra vaagheid in een patent gieten: <http://www.google.com.ar/patents/US8302982>

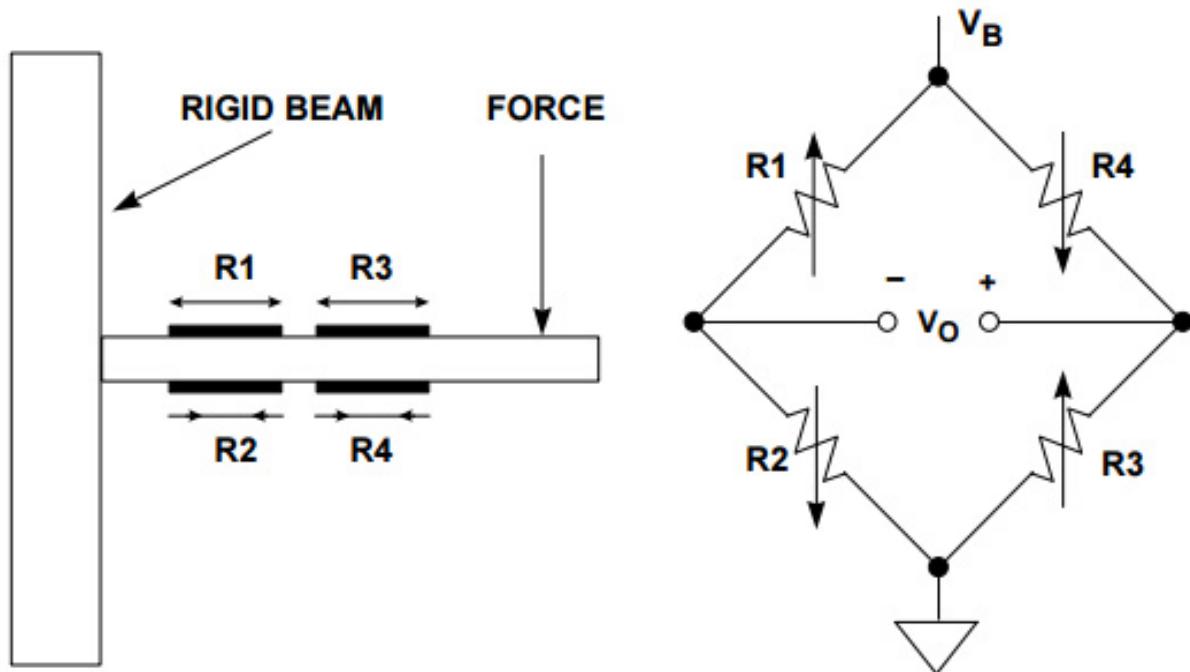
Top. Accel Group, waar Sparta onder valt, heeft nog een hele berg met patenten. Bijv. een patent op vouwfietsen? <http://www.google.com/patents/US6688625>)

In dit geval zit de sensor achter een blobje expoy wat om de as heen zit verstopt, en er komen drie draadjes uit. Geen enkel part nummer, of hint naar wat het precies is. De kans is redelijk groot dat het een of meerdere rekstrookjes zijn (strain gauge).

Een strain gauge, is in feite een weerstand, die van waarde veranderd naarmate je hem verbuigt, verdraait, verrekt of kapot maakt. Omdat de weerstands- waarde maar een heel klein beetje veranderd, is het gebruikelijk er 4 te pakken.

Uit het eerste beste artikel wat ik kon vinden, heb ik ter illustratie het volgende plaatje gejet:

STRAIN GAGE BEAM FORCE SENSOR



De 4 strain gauges, zijn in dit plaatje 4 weerstanden. De configuratie waar ze in staan wordt vaker gebruikt voor allerlij toepassingen en heet een Wheatstone Bridge.

Het voordeel in deze opstelling, is dat de kracht twee strookjes laat uitrekken, en twee inkrimpt. (Andersom als je de kracht omdraait.) Dit heeft als effect dat het signaal wat we gaan meten, V_O , aan de linkerkant meer naar boven verschuift, en aan de rechter kant meer naar onder. En dat verschil is wat we hier gaan meten.

In dit geval is de kracht die we willen meten ietsje anders. Je wilt niet meten hoeveel de as doorbuigt. (Want als er dan een (doorsnee) Amerikaan op gaat zitten, klopt er gelijk niks meer van.). We willen meten, hoeveel de as verdraait.

Dit klinkt vreemd. Waarom zou een as verdraaien, hij zit toch vast? Dit heb ik zelfs eerder gezegd:

- Je moet alles vast willen maken.

Ja. Maar zelfs als het vast zit, verdraait hij een klein beetje als je er kracht op zet. Een miniscuul beetje, en dat kun je dus meten.

Het eerder genoemde blobje epoxy had dus drie aansluitingen. De weerstand van de drie meet-permutaties zijn 350, 350 en 700 Ohm. (Het zijn dus waarschijnlijk 2, i.p.v. 4 rekstrookjes.)

Mijn gedachte was dat als je trapt, die waardes naar bijv. 340 en 360 Ohm veranderen. Als je een vaste spanning op de serie zet, en het midden uit meet, meet je verandering. En moment dat je verandering meet kun je dat als signaal gebruiken om als een idioot weg rijden. Simpel.

Dit moet getest worden. Ik gaf in de conclusie in een van de vorige posts aan dat de huidige test methode sub-optimaal was. Daar moest voordat ik hier mee verder ging, eerst verandering in komen:

De permanente testopstelling.

Dit maakt het leven zoveel makkelijker



Het zijn twee motoren, met een band er tussen gespannen. De linker mist al zijn ingewanden. Zodat er direct iets aan de windingen gehangen kan worden. De rechter zit de vervangende print in, met de originele 6-fase brug.

Ik vind, naast dat dit de meest praktische testopstelling is die ik tot nu toe gemaakt heb, het ook lollig als vensterbank ornament.

Als je je verveelt, kun je het gewoon niet laten om er even aan te draaien...

Omdat ze aan elkaar vast zitten, kan de ene als motor werken, en de ander als generator. Ik kan er nu dingen doen zoals een lamp er aan hangen, of je kunt ze laten touwtje laten trekken door ze in tegengestelde richting te zetten.

Om het leven extra eenvoudig te maken, heb ik de helft van de as waar de draadjes uit komen uitgeboord, zodat er nog meer draadjes uit kunnen. Bij de linker zijn dat de hall sensoren, bij de rechter een programmeer kabel, en aansluiting voor het display.

In de testopstelling (links) zit de vervanging met alle test draadjes naar buiten gefriemeld, in de rijdende testopstelling (rechts) zit het wel netjes:



Zo. We waren aan het meten.

Er kwam in eerste instantie een signaal uit, maar wat ik ook deed, het veranderde niet echt. Ik heb alle motoren als defect gekocht. Misschien zijn de sensoren wel defect? Dat kan natuurlijk.

Ik ben toen eerst maar wat gaan lezen. Met behulp van het eerste beste artikeltje, met alternatieve spelling voor gauge.

...A typical strain-gage based load cell

bridge will have (typically) a 350W impedance.

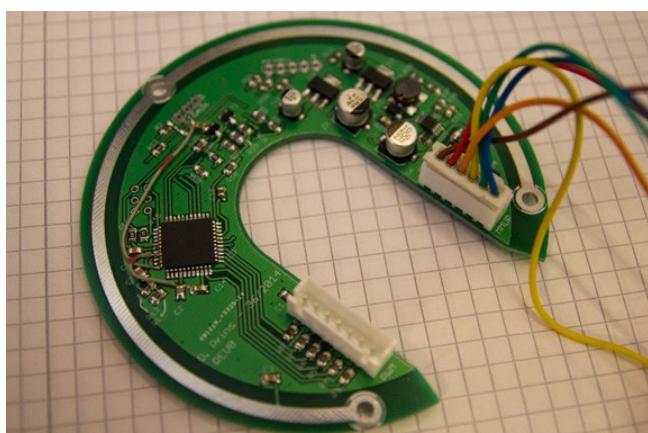
Kennelijk is die van mij dus vrij typisch dan.

Een beetje lezen leert dat naar alle waarschijnlijkheid de volle kracht meting een signaal in de orde van 10mV op levert. Dat is in hun voorbeeld met 10V voeding. In ons geval is dat minder dan 1 Volt. Om een beetje kracht uit te oefenen kan ik moeilijk op de as gaan staan terwijl ik op mijn toetsenbord bezig ben, dus er zal bij mij nog wel minder dan 1mV uit gaan komen.

Er zit ook een mooi schemaatje bij, met allemaal rammend dure onderdelen, waar ik niet echt zin in had. Het is ook niet echt nodig. Ik hoef geen absolute nauwkeurigheid, als het signaal stabiel is en een zinnige verandering geeft als ik trap, kan het met een beetje software voodoo prima gaan werken.

Om een differentiële meting te doen, heeft deze microcontroller intern een opamp. Die kan zonder verdere poespas exact voor dit doeleind gebruikt kan worden.

D.m.v. een extra weerstandsdeling kan deze ook in een Wheatstone schakeling uitgelezen worden, het vereiste even wat ge-tweak:



Bodge wire.

Er komt nu na voldoende versterking wel een signaal uit.

In de firmware gaat het signaal effectief door een compressor heen. Dit levert het uiteindelijke trapsignaal op.

Het signaal moet boven een bepaalde grenswaarde komen voordat het doorgelaten wordt. Deze grens is via het display in te stellen. Vervolgens kan dat nog een keer versterkt worden, zodat je kan instellen of je veel of weinig kracht moeten zetten.

De piek van dit signaal wordt een tijdje vast gehouden (omdat de kracht die je zet op de trappers pulserend gebeurt) alvorens het terug mag lopen.

Ik kan nu rustig tegen de as duwen, en dan begint hij te draaien: [youtube](#)

Github

Omdat er een aantal mensen deze test printjes gaan ontvangen, en omdat het eigenlijk altijd al zo had moeten zijn, heb ik de hele handel op github gezet.

Alle Eagle files, de BOM lijst, de firmware source code, de bootloader en bijbehorende software staat in dit project, en was ik van plan daar bij te houden. Er is zelfs documentatie en een handleiding, met als doel dat als je ook zo'n ding wilt hebben om een wrak mee tot leven te wekken, dat dan relatief eenvoudig zou moeten kunnen.

Alles is vrij gegeven onder GPLv3 licentie. Dit is een prachtige licentie die zeer kort samengevat inhoud: Doe ermee wat je wilt.

Het testen gaat nu vrij praktisch. Ik heb een bootloader geschreven zodat je nu kan nu een kabel tussen de motor en display kan prikken, waarover een programma dan nieuwe firmware de motor in kan schieten. Zo hoef ik niks meer uit elkaar te halen. Ideaal.

Dus, als jij nu bijv. zoiets hebt van: "Ooh, dit wil ik ook. Maar ik wil een grote wijzer op de motor plakken, en er een klok van bouwen." Dan kan dat. Voor vragen mag je me altijd mailen.

Door alles in te bouwen in de motor, is ook de hoeveelheid duc(t)tape tot een minimum gereduceerd:



Ik blijf er lekker rondjes mee rijden, en testen. Op de todo lijst staat nog:

- Verlichting, rem en gas inputs hieraan koppelen.

Verlichting zat oorspronkelijk op de accu aangesloten, die maakte 6 volt zodat er een standaard fiets lamp op kan.

- Iets maken wat de lader uit gooit voordat de accu zichzelf in de brand zet.

Dat deed oorspronkelijk de accu ook. De originele lader maakt idioot veel herrie, ik denk dat er iets moet komen zodat er een laptop adapter gebruikt kan worden. Op het moment moet ik hem zelf uit doen als hij warm begint te worden.

Verder begint het op te vallen, dat sommige honden intens graag in de motor willen bijten als ik voorbij rijd... Is er wellicht een expert die mij kan vertellen welke frequentie honden irritant vinden?