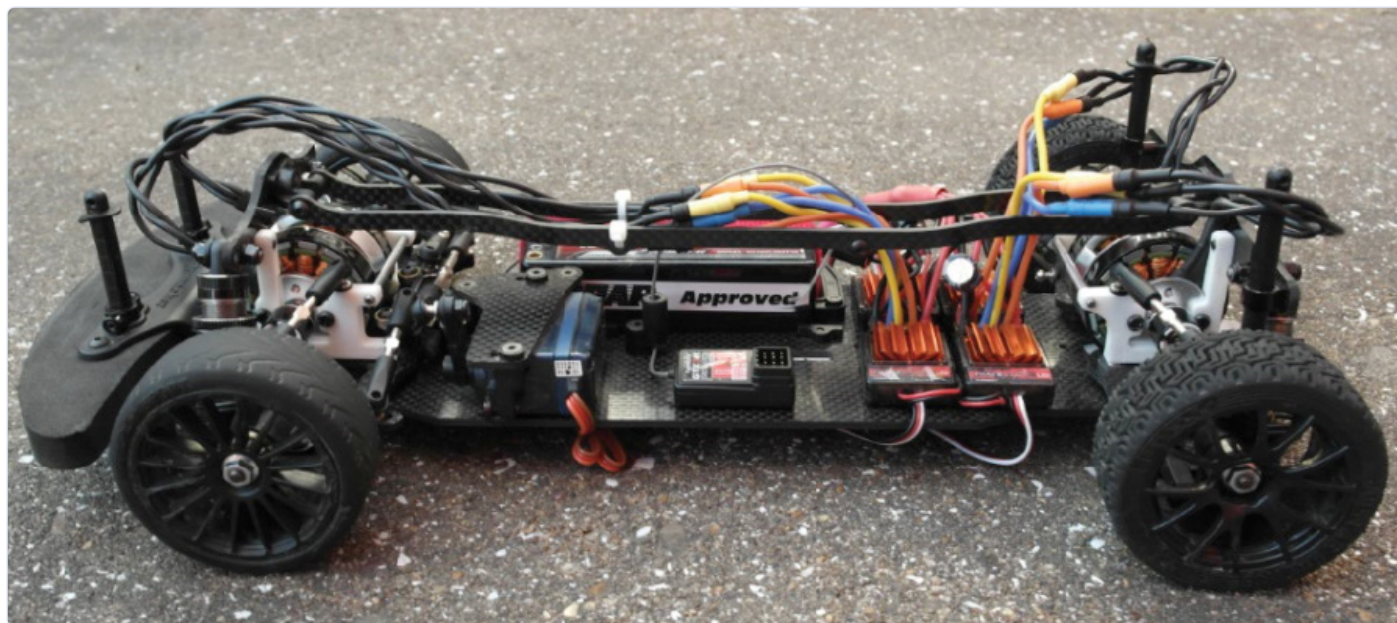


Error analysis

tips en trucs: FMEA, grote stromen en meer

samengesteld door **C.J. Abate**

Overmatige elektromagnetische interferentie (EMI), een gebraden print, een geëxplodeerde batterij en verkeerde sensorwaarden: dat alles kan het gevolg zijn van een eenvoudige ontwerpfout. Om een betere professionele ingenieur of serieuze elektronica-maker te worden moet u van uw ontwerpfouten leren (en ook van andermans m/v fouten). In deze nieuwe artikelreeks “Error Analysis” delen we inzichten van leden van de Elektor-community over hun ontwerpfouten en wat ze daarvan hebben geleerd. En natuurlijk wijzen we daarbij op tips en trucs die u kunnen helpen uw ontwerp-, test- en programmeervaardigheden te verbeteren.



Figuur 1. Het RC-model van Rajesh Nakarja.

Verbeter de betrouwbaarheid van uw ontwerp met FMEA

Het maakt niet uit hoe ingewikkeld een project is, toepassing van Failure Mode and Effects Analysis (FMEA) kan het verschil betekenen tussen succes en catastrofe. Rajesh Nakarja, een ingenieur uit Stockholm (Zweden), weet dat maar al te goed. Hier deelt hij het verhaal van een problematisch Arduino-gebaseerd RC-ontwerp en, natuurlijk, wat hij daarvan heeft geleerd:

“Als student was ik dol op RC-modellen. Mijn diploma destijds, in embedded control, was aanleiding om een eigen DSP-controller te ontwikkelen

voor koppelverdeling bij radiobestuurde auto's op schaal 1:10. Het voertuig, met vier onafhankelijk aangedreven wielen en meerdere sensoren, kon extreme snelheden bereiken die normaal gesproken gereserveerd zijn voor racewedstrijden op een echt circuit (**figuur 1**). Met *real-time processing* bleef het voertuig echter ook bij topsnelheid stabiel dankzij detectie van tractieverlies en automatische correctie met gyroscopen.

De code zelf is geïmplementeerd vanuit de Simulink-coder op een aangepast STM32F4-board en bevatte tal van software-beveiligings-features. Over-/underflow, verlies van radiocontact of sensorgegevens, voor dat alles bestond een veilige oplossing zodat het voertuig tot stilstand zou komen. Veel tests en simulaties toonden aan dat



Moderne tools en methoden, zoals modelgebaseerd ontwerp, zorgen voor omvattende automatische tests in de ontwerpfase.

dit goed werkte en ik had er alle vertrouwen in dat het de auto in de praktijk veilig zou zijn.

Met een gewicht van een paar kilogram zorgden de snelheid en het vermogen van vier 40-A BLDC-motoren ervoor dat crashes behoorlijk hard konden aankomen voor mens, dier of wat dan ook. Als gevolg hiervan was ik buitenshuis heel voorzichtig en wilde ik het RC-model niet in de buurt van mensen of verkeer gebruiken.

Voor de besturing had ik een Xbox-controller met een Arduino-Zig-Bee bridge gecombineerd; deze zorgde voor real-time besturing en snelheidsregeling. Hoewel de autocontroller zelf betrouwbaar en robuust was ontworpen, was de Arduino-controller meer een soort

het zicht was. Terwijl ik probeerde de batterijen van de Arduino los te koppelen, vloog het voertuig langs een groep mensen en knalde toen tegen een stoeprand met misschien wel 50 km/u. Gelukkig raakte niemand gewond; de kracht van de botsing had tot gevolg dat de lithiumbatterij uit het chassis werd gescheurd zodat de stroom uitviel. Beschaamd en sorry roepend rende ik naar het vernielde model en ging ik met wijlen een project naar huis (**figuur 2**).

Wat ik die dag leerde was dat het kleinste foutje voor de andere veiligheidsvoorzieningen in een systeem compleet normaal kan lijken. Het maakt niet uit hoe ze zijn ontworpen – een enkel klein foutje kan voldoende zijn om een heel systeem te doen crashen. Sindsdien heb

ik de analyse van mogelijke foutmechanismen veel serieuzer genomen en pas ik die in principe vanaf het eerste begin toe op elk project waar ik aan werk.

Moderne tools en methoden, zoals modelgebaseerd ontwerp, zorgen voor omvattende automatische tests in de ontwerpfase. In combinatie met een evoluerend releaseplan betekent dit dat *bugs* al vroegtijdig worden voorkomen of in het releaseproces snel worden opgemerkt. Hoewel er nog altijd iets mis kan gaan, betekent een goed testplan dat iets eigenlijk nooit meer dan één keer mis gaat. De foto's tonen het door mij gebouwde model met besturingsprint. Alle elektronica is gecoat om schade door vuil of vocht te voorkomen. Een video op YouTube beschrijft het project in meer detail [1]."



Figuur 2. Soms gaat het goed mis!

hack; de enig toegevoegde beveiliging was dat bij verlies van radio-contact de auto zou worden uitgeschakeld. Dat leek toen voor een eenvoudige controller een redelijke veiligheidsvoorziening.

Toen ik op een middag de auto uitprobeerde in een vrij toegankelijk park, presteerde de USB-controller het om kortstondig het contact met de Arduino/ZigBee-eenheid te verliezen. Hierdoor crashte de USB-driver en bleef de radio een 'plankgas'-commando uitzenden – met als gevolg dat het voertuig op volle snelheid koers zette naar drukke straat en hoofdweg.

Ik had Zigbee-modules voor communicatie over grote afstand gebruikt, zodat de beveiliging pas zou ingrijpen als het model ruimschoots uit

Houd grote stromen onder controle!

Lars Krüger is een leraar uit Potsdam (Duitsland) die al acht jaar Elektor leest en sinds 1992 elektrische fietsen ontwerpt. U kunt zich voorstellen dat hij in de loop der jaren veel geleerd heeft over het regelen van grote stromen, het testen van schakelingen en e-bikes:

"Vroeger, in mijn jeugd, wilde ik dingen zo snel mogelijk af hebben. Een snel idee in de ochtend moest 's avonds realiteit zijn. Een paar storingen interesseerden me niet als de rest redelijk in orde was en werkte. Maar dat leidde tot veel kapotte en verbrande halfgeleiders als het hogere spanningen en/of grotere stromen betrof!

Zo bouwde ik in 1992 mijn eerste elektrische fiets. Ik had nog niet de helft van een boek over die materie gelezen toen ik al een PMW op ongeveer 100 Hz had gebouwd, in combinatie met een setje van 10 'BUZ'-MOSFET's, een vette 100Ah-autoaccu en de 2kW-startmotor van een Citroen CX 2.5D. Na de eerste maal gas geven bleef de motor



Figuur 3. De e-bike-projecten van Lars Krüger leerden hem het een en ander over het regelen van grote stromen. Dit is zijn e-bike uit 1994.

'aan' (de MOSFET's waren verbrand) en de fiets, die een snelheid van zo'n 60 km per uur bereikte, was op hol geslagen.

Uitschakelen van de voedingsspanning van de PWM bracht geen soelaas omdat de verbrande FET's zich niet meer voor hun gate-spenning interesseerden. Gelukkig had ik wel een zekering ingebouwd; het betere remwerk leide ertoe dat deze doorbrandde. En daarom construeerde ik een 'zware' schakelaar van dik koper en een veer. Deze werd bediend met een Bowdenkabel in plaats van met een potentiometer zoals bij het eerste experiment. In deze 'onbedorven' vorm was de fiets dikke pret – 10 meter trappen en dan 'zzzoef'. In de smalle straatjes van mijn geboortestad kon ik de fiets niet onder controle houden, maar daarbuiten was het een puur genoegen.

Wat ik uit dit verhaal heb geleerd is dat het regelen (begrenzen) van stroom een noodzaak is; wanneer u een zwaardere motor gaat gebruiken, mag u dat beslist niet vergeten. En ook dat deze regeling snel genoeg moet zijn om een snelle stroomtoename aan te kunnen, anders gaat de regellus oscilleren. Later bestudeerde ik 'regeltechniek', al dat 's+1'-gedoe met gering effect. Dat had weinig met het echte leven te maken. Ik heb meer geleerd van simulatietools zoals PSpice of zijn opvolger SIMetrix, die toen in opkomst waren. Daarna had ik veel minder problemen met het regelen van grote stromen. Maar natuurlijk brandt er in mijn thuislab zo af en toe nog altijd een MOSFET door. Soms glijdt een probe weg met kortsluiting tot gevolg. Goed wanneer u dan een veiligheidsbril draagt en de verlichting van het lab op een andere groep is aangesloten. De foto toont overigens mijn tweede e-bike uit 1994 (figuur 3). De eerste was zo snel dat het me niet lukte er een foto van te maken..."

MOSFET's, BJT's en PLL-regeling

Robert Owen is Senior System Engineer uit Stenløse (Denemarken), met jarenlange ervaring op verschillende gebieden van elektronica. Hier beschrijft hij wat hij heeft geleerd tijdens het ontwerpen van een sync-separator en -tracker voor CRT-gebaseerde videoprojectoren. Nadat hij een schakeling met een MOSFET had opgebouwd, werd hij

DEEL UW FOUTEN MET ONS!

Wilt u bijzonderheden over uw elektronische vergissingen en wat u daarvan leerde met ons delen? Vul dan ons "Error Analysis"-formulier in. Bijdrage van professionele elektronici, makers en studenten zijn meer dan welkom.

www.elektormagazine.com/pages/error-analysis-submission

al snel met een probleem geconfronteerd. Gelukkig bleek dat eenvoudig op te lossen, en leerde hij het een en ander over MOSFET's, BJT's en PLL-regeling:

"Vele jaren geleden, in een ander leven, was ik betrokken bij het ontwerpen van de synchronisatiescheiding en trackingcontrole voor CRT-gebaseerde videoprojectoren. Om tegemoet te komen aan gebruikers die eisten dat hun goedkope videorecorder een perfect beeld gaf met onze peperdure projectoren, moesten we de PLL-regellus kunnen onderbreken tijdens het omschakelaarinterval van de koppen van de videorecorder. De controle-chip had een ingang voor lusbesturing die bruikbaar zou moeten zijn. Dus monteerde ik een MOSFET-transistor als variabele weerstand en bouwde een nieuw board. Maar de schakeling onderbrak niet alleen de PLL-lus niet, maar het probeerde het sync-sigitaal nog nauwkeuriger te volgen, resulterend in scheeftrekken van het beeld aan de bovenzijde van het scherm. (Onze projectoren toonden alle beeldlijnen, in tegenstelling tot een TV-toestel dat 10 lijnen of zo aan beide kanten afkapt.) Ik ontdekte dat de stuurgang reageerde op de stroominjectie via de zeer grote gate/source-capaciteit. Vervanging van de MOSFET door een JFET, of in ons geval een eenvoudige BJT, loste het probleem op zonder dat de print moest worden aangepast." ❏

200303-04

WEBLINK

- [1] "Independent Wheel Drive 1/10 Scale RC Car - Project Aelith", SiliconWitch Couture, YouTube, 2014 : <https://youtu.be/ET1HEyqEQJQ>