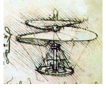


Toyota's killer firmware: Bad design and its consequences

Michael Dunn -October 28, 2013

109 Comments

Share 269 g+1 916 Tweet 707 Like 3.7k

On Thursday October 24, 2013, an Oklahoma court **ruled against Toyota** in a case of unintended acceleration that lead to the death of one the occupants. Central to the trial was the Engine Control Module's (ECM) firmware.

Embedded software used to be low-level code we'd bang together using C or assembler. These days, even a relatively straightforward, albeit critical, task like throttle control is likely to use a sophisticated RTOS and tens of thousands of lines of code.

With all this sophistication, standards and practices for design, coding, and testing become paramount – especially when the function involved is safety-critical. Failure is not an option. It is something to be contained and benign.

So what happens when an automaker decides to wing it and play by their own rules? To disregard the rigorous standards, best practices, and checks and balances required of such software (and hardware) design? People are killed, reputations ruined, and billions of dollars are paid out. That's what happens. Here's the story of some software that arguably never should have been.

For the bulk of this research, EDN consulted Michael Barr, CTO and co-founder of **Barr Group**, an embedded systems consulting firm, last week. As a primary expert witness for the plaintiffs, the in-depth analysis conducted by Barr and his colleagues illuminates a shameful example of software design and development, and provides a cautionary tale to all involved in safety-critical development, whether that be for automotive, medical, aerospace, or anywhere else where failure is not tolerable. Barr is an experienced developer, consultant, former professor, editor, **blogger**, and **author**.

Barr's ultimate conclusions were that:

- Toyota's electronic throttle control system (ETCS) source code is of unreasonable quality.
- Toyota's source code is defective and contains bugs, including bugs that can cause unintended acceleration (UA).
- Code-quality metrics predict presence of additional bugs.

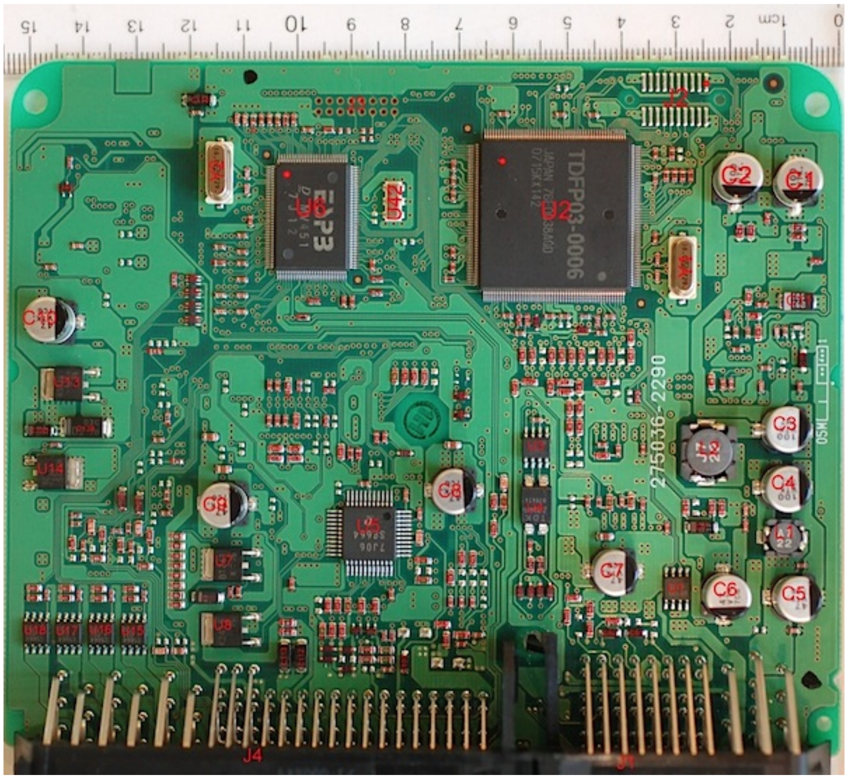
- Toyota’s fail safes are defective and inadequate (referring to them as a *“house of cards” safety architecture).*
- Misbehaviors of Toyota’s ETCS are a cause of UA.

A damning summary to say the least. Let's look at what lead him to these conclusions:

Hardware

Although the investigation focused almost entirely on software, there is at least one HW factor: Toyota claimed the 2005 Camry's main CPU had error detecting and correcting (EDAC) RAM. It didn't. EDAC, or at least parity RAM, is relatively easy and low-cost insurance for safety-critical systems.

Other cases of throttle malfunction have been linked to tin whiskers in the accelerator pedal sensor. This does not seem to have been the case here.



The Camry ECM board. U2 is a NEC (now Renesas) V850 microcontroller.

Software

The ECM software formed the core of the technical investigation. What follows is a list of the key findings.

Mirroring (where key data is written to redundant variables) was not always done. This gains extra significance in light of …

Stack overflow. Toyota claimed only 41% of the allocated stack space was being used. Barr's investigation showed that 94% was closer to the truth. On top of that, stack-killing, **MISRA-C** rule-violating recursion was found in the code, and the CPU doesn't incorporate memory protection to guard against stack overflow.

Two key items were not mirrored: The RTOS' critical internal data structures; and—the most important bytes of all, the final result of all this firmware—the TargetThrottleAngle global variable.

Although Toyota had performed a stack analysis, Barr concluded the automaker had completely botched it. Toyota missed some of the calls made via pointer, missed stack usage by library and assembly functions (about 350 in total), and missed RTOS use during task switching. They also failed to perform run-time stack monitoring.

Toyota's ETCS used a version of **OSEK**, which is an automotive standard RTOS API. For some reason, though, the CPU vendor-supplied version was not certified compliant.

Unintentional RTOS task shutdown was heavily investigated as a potential source of the UA. As single bits in memory control each task, corruption due to HW or SW faults will suspend needed tasks or start unwanted ones. Vehicle tests confirmed that one particular dead task would result in loss of throttle control, and that the driver might have to *fully remove their foot from the brake during an unintended acceleration event* before being able to end the unwanted acceleration.

A litany of other faults were found in the code, including buffer overflow, unsafe casting, and race conditions between tasks.

Share 269 g+1 916 Tweet 707 Like 3.7k

< Previous Page 1 of 2 Next >

109 Comments

Write a Comment

To comment
please Log In

djhk

Let's get real. Software is not perfect. Murphy says that if it can fail it will.
A driver needs a simple way to stop a car that does not include any CPU.
Large trucks have a Fuel Pump cut off switch that will stop the truck.
Cars need the same safety feature.

Dec 26, 2013 12:44 PM EST

Reply

Lilah

Um yeah Toyota got schooled. It's just unfortunate that a life was lost in the process and Toyota should have to face those consequences. Perhaps this a good lesson for Toyota to remember - there's always a source code expert out there that is smarter than you. I used a source code expert witness from **http://ironwoodexperts.com/** in a case once and they did a phenomenal job so I know firsthand how smart this type of expert is.

Nov 12, 2013 1:48 PM EST

Reply

Michael Dunn

Missing comments?

According to my notification emails, on Oct 29 between 4 & 5PM ET, 7 comments were posted which didn't show up onsite. My apologies for the hiccup. Feel free to repost your comment if it was one of those seven!

Thanks.

Nov 11, 2013 11:09 AM EST

Reply

harpat

Software controlled throttle was an inherently bad idea. It does not matter how clever the programmers are. The problem is that there are always failure modes that are difficult or impossible to reproduce. Often, you have to know what caused the failure in order to test for that failure mode. When I worked at laptop companies, I designed battery chargers and was responsible for the charging architecture and for years I designed charging controls with hardware and had independent safety circuits. For the most parts, things went well but one time a prototype laptop battery caught fire at the most inopportune when a major OEM customer was evaluating it. We lost that customer. Our battery pack supplier quickly identified the problem as that due to electrolytic leakage from a mis-installed seal.
I am glad that I had resisted all attempts from colleagues, management and influential outsiders to incorporate software controlled charging and Intel was pushing smart battery technology which went against my grain. If a failure were caused by software bug, both my company and my personal career would be toast. When finally our management succumbed to Smart battery propaganda pressure, I excused myself from that responsibility. To make a long story short, so many strange bugs appeared in the development, we lost the marketing window and the development group became toast. A few years latter Intel threw in the towel on Smart battery technology. My motto in life is, use digital technology for information and entertainment but not for mission critical operations when it can be avoided. After all, the blue screen of

death has never been eliminated.

Nov 6, 2013 6:24 PM EST

Reply

Michael Dunn

How do you propose to implement electric vehicles then? Fully hardware based? Might be doable...

Nov 6, 2013 8:08 PM EDT

Reply

Antony Anderson

In a gasoline engine powered vehicle the power stage of the electronic throttle control might be rated at a peak power of 10-20 watts. In the Tesla model S the variable speed AC induction motor will be driven by a three phase inverter capable of delivering 310 kW. The power ratio of the two drives is of the order of 15,000 to one. This gives you an idea of the difference in scale of the switching electronics used in each case that require fault protection. I am sure the Tesla motor company would adopt a similar protection and isolation strategy as would be provided for a stationary variable speed drive of the same size. The battery in the Tesla S has 7000 cells. I do not know how they deal with charging evenly and detecting and isolating battery faults. I expect too that Tesla will be needing to incorporate some MALware protection before long because Stuxnet will be coming and causing the motor to overspeed as per the Iranian centrifuges.

Nov 7, 2013 3:35 PM EST

Reply

harpat

It should be possible to do it in hardware. Software may have some role in monitoring the status of the battery but even there pitfalls exist. For example, if a false warning of depleted battery is issued while you are driving in the boondocks, you may get stranded if you heed those warnings. So there should be an easy to use hardware monitor for emergencies. I have experienced many laptop shutdowns due to software determining incorrectly that the battery is depleted. Unfortunately, the more popular Li-ion cells have a flat discharge profile which force you to use software integration of current (coulomb counting) to determine remaining capacity. Invariably the counting drifts and requires a complete charge discharge cycle to reset the counter and it is not convenient for the user. The result is probably millions of battery packs are discarded or returned unnecessarily. I preferred a cell chemistry that had a sloping discharge curve that enabled approximate remaining capacity so it prevented the gross error drift of coulomb counters. The trade off was about 10% less energy storage per unit volume. The way I looked at it is the higher capacity is of no use if software prematurely shuts down the system every so often.

Dec 19, 2013 4:20 PM EST

Reply

Michael Dunn

One of my favourite quotes is, I think, very applicable to mission-critical software:

"Everything should be as simple as possible, but not simpler."

(ascribed to Einstein)

Nov 5, 2013 4:48 PM EST

Reply

Michael Dunn

BTW, there is an online "chat" (text only) on Wed Nov 6 at 1000PT, 1300ET, 1800UTC focusing on Toyota UA, at:

http://www.eetimes.com/messages.asp?piddl_msgthreadid=43358

I'll be there...

Nov 5, 2013 4:41 PM EST

Reply

theoldwizard1

I worked directly and indirectly developing embedded automotive powertrain control software for 30+ until 2007. In the "early days" everything was hand coded assembly language and the ROM size meant that it was not difficult for the software engineers to actually understand all of the code.

By the time 32 processors with high level languages became the norm, it was well beyond the capability of one person to understand a significant part of massively complicated code or even the complex interaction between control subsystems.

Sadly, lower, middle and upper level management have no comprehension of any of the issues discussed in the article. I saw arbitrary changes in CPU architecture in the middle of a complex design with only a few weeks added to the schedule. The reality was several months and a lot of overtime !

I can only speak from my personal experience, but zero percent of the management (literally hundreds) had ANY formal education/degree in software engineering. Upper management feels that software development is like making a spreadsheet to add up a few columns of numbers.

Nov 5, 2013 2:48 AM EST

Reply

Michael Dunn

Re tin whiskers, isn't *automotive* critical enough to get a RoHS exemption? Yeesh. Swatch watches got one!

Nov 4, 2013 6:24 PM EST

Reply

BB gun

Michael,
From some anecdotal evidence, it appears that automakers are not even trying to get an exemption. They are just trying to make the best of it. (I realize that solder and connector pins are two different issues, but) I asked one of Toyota's major suppliers of soldering equipment whether

the automakers plan to fight for whiskers reducing tin-lead in their applications, and the technical staff there said no, they are just trying to cope with RoHS. I asked how they plan to mitigate tin whiskers risk. "More research is needed." Hmm.
I wonder what consumers are supposed to do in the meantime.
Can anyone here comment on whether this impression is accurate?

Nov 5, 2013 12:08 AM EDT

Reply

BB gun

During the long pause between the NASA study of Toyota's electronics and the revelations in the Oklahoma trial, there were a few media reports that raised questions about Toyota's position.
Toyota's top lawyer wrote a threatening letter to CNN before it aired a SUA related story that deserves to be revisited now.
http://i2.cdn.turner.com/cnn/2012/images/02/26/toyota.2-22-2012.letter.to.cnn.pdf
Pretty chilling.
After another news report, Toyota's top PR guy had this to say. 'There are no real-world scenarios in which Toyota electronics can cause unintended acceleration."
Read it and weep:
http://www.huffingtonpost.com/mike-michels/tin-whiskers-and-other-di_b_1231080.html
That was in response to a leaked memo by a Toyota executive who was warning his colleagues that their cover story will inevitably unravel under the scrutiny of litigation.

Nov 2, 2013 7:11 AM EST

Reply

BB gun

The ECM board may have other hardware-related problems not noted by Barr. Dr. Michael Pecht, a microelectronics reliability expert who is familiar with Toyota SUA issues, examined the connector pins on a Toyota Truck ECU, found them to be pure tin, and then grew some tin whiskers on them in his lab at U of

Maryland CALCE. An abstract of his findings can be found here:

http://www.sciencedirect.com/science/article/pii/S0026271413003107

While the presence of tin whiskers in the ECM is not proved to lead to UA, (although this is an ECM from a truck model that was subject to alleged SUA in a settled case) they could be responsible for electrical chaos that the firmware failsafes may or may not detect. I'd be interested in comments from engineers on how tin whiskers among ECM connectors might affect vehicle behavior in an ECM that is running such buggy software.

Nov 2, 2013 6:56 AM EST

Reply

Antony Anderson

Re Tin whiskers and potential knock on effects:

"Great fleas have little fleas upon their backs to bite 'em,
And little fleas have lesser fleas, and so ad infinitum.
And the great fleas themselves, in turn, have greater fleas to go on,
While these again have greater still, and greater still, and so on."

It seems to me that a tin whisker could cause a transient short circuit and local voltage dip which might cause a cause a cascade of things to happen, resulting eventually in a sudden acceleration. If a tin whisker could cause an intermittent microphonic contact then mechanical vibration could be converted into electrical noise which might set off no end of trouble. It might be useful to think in terms of a control system that was very near its stability limit that the tin whisker might push into instability.
I would suggest that monitoring the duty cycle of the PWM controller of the throttle motormight give a very good indication of incipient instability.

Nov 5, 2013 1:21 PM EST

Reply

MrPWM

I have worked with several programmers who have written support code for the analog circuits which I've designed. In most of these cases, I have had to tell them, "slow down, why are you adding all this code which isn't needed?" The reply is always something like, "memory is free so we choose to add a lot of complex stuff just 'cause we can". For example, if a parameter is only required to have 20% accuracy, why are we introducing interpolation between each sample point, then calculating the result to 16 bit accuracy? I wanted 10 lines of code. I got a thousand.
Another example: For closed loop control of a motor drive, the software I had to work with was monitoring so much stuff that I couldn't get the calculations done before the next sample time. Many of these were parameters within an inner loop, which could have been completely ignored since the outer loop corrected for these. Remember open loop gain G/(1+GH)? G can vary by a large amount and the final response will still be ~1/H. You don't need zero point zero zero zero 1 percent accuracy to calculate G. The programmer didn't know this because he had never taken a feedback theory course, or he forgot it.
I'm thinking of the reply to my question of why it takes tens of thousands of lines of code to read the position of a gas pedal. Rathbun explained to us all the items he has to put into his code. He's probably a pretty smart guy and does what the systems engineers want but, is this all necessary? Are the systems engineers "over designing" it?

Nov 1, 2013 3:11 PM EST

Reply

Oldandgray

Many years ago I had a ’72 Torino with a 351CJ engine. There was a problem with the automatic choke linkage that could stick the throttle wide open when the accelerator was floored.

One time, I was entering the freeway from a ramp that merged into the left lane. It was necessary to cross 3 traffic lanes within ¼ mile in order to exit the freeway on the right. At the end of the on ramp I floored the accelerator and the throttle stuck wide open. I immediately shut off the ignition. This caused the steering wheel to lock so I had no steering. I turned the ignition part way back on to regain steering and pulled over to the side of the road safely.

The point is that mechanical systems can also fail. Cars today are way better than they used to be.

Nov 1, 2013 11:40 AM EST

Reply

harpat

Yes, mechanical systems can fail as well and it is important to have two or more robust safety interruption devices (not flimsy signalling devices for emergency.) Elevators typically have several back up devices. They are suspended on multiple ropes and if they all brake, the elevator will instantly stop because 4 dogs will grab the rails when rope tension is lost.
The most serious problem with software control is there are usually too many ghosts lurking around and you can never be sure if you caught any or all of them. So if you have million cars on the road and the production lines are running full blast when these ghost problems occur, you have an incredibly difficult situation because you cannot identify the problem decisively. Mechanical problems usually are easier to identify and corrective actions taken. Apparently you knew you had a sticky throttle and you ignored it rather than have it fixed. Luckily, the ignition switch worked for you but perhaps a better additional safety device should cut off the fuel supply when the brake pedal is floored with significant force. Losing engine control can be of course hazardous depending on traffic conditions conditions.

Dec 26, 2013 7:49 PM EST

Reply

rberra

Please change "Mr. Berra" to "Mr. Barr" in my previous comment.

Oct 31, 2013 11:35 PM EDT

Reply

rberra

In his testimony during the Bookout trial, Mr. Berra stated that Toyota’s software expert, Mr. Ashish Arora from Exponent Inc, wrote a report on September 17, 2012 in which he gave his test results on Task X death in a later model Camry (2008 MY) and its association with unintended acceleration. In the report Mr. Arora states that during the test when he was pressing the brake during Task X death, the recorded black box data sequence said he didn’t press the brake. This statement casts doubt upon all EDR brake data obtained during a sudden unintended acceleration incident in a Toyota vehicle.

Oct 31, 2013 11:32 PM EDT

Reply

ShaneBK

I would be interested in seeing similar reports from a few other manufacturers controllers before condemning Toyota out-of-hand. There is always an ignition switch ... assuming it is not a diesel running on it's own lube' oil. It takes a few seconds, but unless it is a high performance car, it won't gather THAT much speed in that few seconds. Hazardous to pedestrians, but not necessarilly other vehicles ... if the driver keeps his/her head.

Oct 31, 2013 2:33 PM EDT

Reply

bearslumber

Hi ShaneBK,

It is hazards to people (pedestrians and passengers alike) that we primarily protect in safety critical software. You can replace a vehicle but you can't replace a persons life.

The report is still just as damning.

Nov 2, 2013 12:06 PM EST

Reply

rbeckmann

Yes there is, a button wired right back into the same ECU. Also pushing the STOP button will not cut the engine unless car is in park and not moving. Another brilliant idea. Assuming the CPU and OS were still processing anything you need to hold the button for about 6 seconds to kill the engine. Something I doubt anyone knew at the time. My guess is engineers will think more along the lines of a power killing fail save system just like we have on our laptops if the hang.

Nov 20, 2013 8:31 AM EST

Reply

Richard_Brabant

I read all your "excellent engineer savvy comments" but I feel there is a huge problem that is barely touch under the paragraph: resolutions; and the open questions at the end of the article.
The industry integrated more and more electronic and software complexity into everything.
The cost reduction and pressure for delivering faster to the market are totally opposite to increase complexity of the embedded issues!
The dynamic of the competition in the market is to work with "GOOD ENOUGH".
How many very experienced engineers were exited as too expensive and also slowing down the project schedule?
The technical managers are facing so much pressure they need to compromise.
The technical managers are asked to know a lot more and to cover so much, many times they cannot get the proper knowledge for taking the right decision!
Also the political decision of eliminating the "no-no" by the "yes-yes" through the all hierarchy of command is pushing the industry farther into the risky zone.
In the technical report we can discover so many bad practices and so light safety reviews and checking! why? the answers are well known: cost, time, wild competition, secrecy.
I really think we are at a turning point in many industries where the complexity and sophistication of the intelligent/smart products are putting too much risk without the needed level of security guaranteed by adequate complying protocols and independent third party certifications.
Ok, I already know my detractors argumentation: today the products are way better than those of yesterday, just so few problems and we do so much.
The trend and correction is needed to be addressed because the future is more complexity, bigger, increase usage and the "designers" restricted to a narrower domain of knowledge.

Richard.

Oct 31, 2013 12:24 PM EDT

Reply

bearslumber

Very good point Richard_Braybunt

However, as Bob Marley once said we can't stop the tide of progress.

We (the safety critical software profession) need to work smarter. While the technology gets more complex, there is a greater demand, and it is our responsibility to step up to the mark and work smarter and at the same time maintain and preferably enhance integrity.

We need to move with the tide otherwise ours is a dying profession.

But I agree with you wholeheartedly that integrity must never be compromised. Costs can only be reduced in exchange for better methods of achieving the same and better levels of integrity.

Nov 2, 2013 12:17 PM EST

Reply

Microchip

It is easy to understand the emotion here, but no one has addressed the question that comes immediately to my mind; is the software development any different in GM, Chrysler, Ford, VW/Audi, FIAT, BMW, Mercedes, Nissan / Renault and all others?
Thinking back to the development of the code for Fly by Wire systems in airplanes, there have been a number of incidents in Aviation that could be attributed to the software and its developers.

Oct 31, 2013 5:28 AM EDT

Reply

bearslumber

That is true Microchip,

But having worked in the industry you mention, I can vouch that lessons have been learned.

Primarily, Independent Safety Authorities (ISA) have been established, and every project must be assessed and passed by the ISA in order to achieve certification. Most of the ISA's that I have worked with have been very dogged and determined and my impression is not much can pass their scrutiny.

My question is why Toyota has achieved certification and how such bad practice has been accepted by any ISA.

Nov 2, 2013 12:25 PM EST

Reply

NIKT

Ye gods, what was so bad about a throttle cable or pair of cables to assure throttle closure.
Maybe my approach to engine mangement is too simplistic, but what does all the electronic "monkey motion" described by all the commenters do that a simple throttle cable and TPS can't do? Just askin' as they say.

OK, so you need an electromechanical servo for cruise control, but is automotive drive by wire worth the risk of UA?

Best regards to all,

Myron Boyajian

Oct 31, 2013 2:14 AM EDT

Reply

skyhooktek

hint " Audi UA" in 60 minutes (cbsnews)

Oct 31, 2013 8:54 PM EDT

Reply

dstechDan

UA = Floormats getting jammed between the bottom of the throttle pedal and the floor board. CBSNews was found guilty of staging the UA proof they showed. This is why almost all OEM's now have anchor points for the driver floor mat to keep it from sliding forward under the throttle pedal. People still create this problem when they put aftermarket mats on top of their OEM mats without anchoring them first. Research the facts.

Nov 1, 2013 8:41 AM EST

Reply

Antony Anderson

Re after market floormats. I have never quite understood the physics of failure mechanism involved whereby a floormat that has been removed from the vehicle can still entangle itself with the throttle pedal and cause a UA. Am I missing something?

Nov 4, 2013 9:49 AM EST

Reply

JR45

We're risking our lives for fuel efficiency :)

Nov 1, 2013 3:53 AM EST

Reply

dstechDan

Exactly, and we also expect the best technology and quicker lifecycle updates as consumers, but yet still want to pay nothing, or a marginal delta, for these lifecycle updates. Cost cutting pressures by the OEM's put on their current suppliers, and the bid wars that go on for new business, ultimately have negative effects on the end product's hardware & software. Basically, I'm not surprised.

Nov 1, 2013 8:52 AM EST

Reply

benmlee2

Simple answer why we need drive by wire: Government regulation. Because of stability control requirement of all new cars, you can't make a car without drive by wire and meet government regulation. Stability control is *suppose* to make a car safer. Of course, like air bags, there are unintended consequence. We have to assume that the end result is a net positive gain in safety.

Nov 1, 2013 10:37 AM EST

Reply

stratus46

I'm on my second Hybrid, Prius then Fusion. The pedal position has exactly nothing to do with the actual throttle position so you have no option to use mechanical throttle linkage.

Nov 2, 2013 3:56 AM EST

Reply

Scuromondo

None of the evidence provided in the article provides sufficient evidence to illustrate which particular software error could have caused the specific reported failures. It is only a long list of instances of bad software practices which, taken together, may create doubt about the overall software quality of the product, but only provide an abundance of circumstantial evidence that the software could conceivably be at fault. It is certainly not conclusive.

Oct 30, 2013 4:30 PM EDT

Reply

jimm1024

I was looking for the smoking gun as well but didn't see it specifically. Although perhaps pointing to an actual repeatable cause (even with one fault condition) could open up a whole other lawsuit from those that tried to recreate it to disastrous end.

Oct 30, 2013 5:26 PM EDT

Reply

skyhooktek

excellent point

Oct 31, 2013 8:52 PM EDT

Reply

benmlee2

Also remember, he is a paid consultant. He is paid to produce damming evidence. I can take any design and make it look horrible by applying all the standards out there. My question is do we really want million dollar Space Shuttle quality throttle in a Camary. Even then, you can still find fault. Realistically, a mission critical component need to go thru acceptance test with environmental test. It will end up costing $100k. What the article did not provide is a balanced view on what is "good enough". The sky is the limit, but what is "acceptable" defect rate. The answer is not zero because if you want zero, then is going to cost you Space Shuttle like $100k or more. That is what the jury does not understand, but we need a balanced view here.

Nov 1, 2013 10:43 AM EST

Reply

LarryOsolkowski

One little detail you're overlooking is that the cost of development for the Space Shuttle was amortized over something less than a dozen units, vs. amortizing over several million Camrys. The cost per unit goes way down.

Nov 1, 2013 3:44 PM EST

Reply

larry.speed ex.mx racer

This was "sufficient evidence" that the S/W "could conceivably be at fault" IMHO:
"Vehicle tests confirmed that one particular dead task would result in loss of throttle control, and that the driver might have to fully remove their foot from the brake during an unintended acceleration event before being able to end the unwanted acceleration."

Nov 2, 2013 4:12 PM EST

Reply

Ken Freeman

"Pedal," not "peddle." A gas pedal is conceptually at least a very simple function. To peddle, or sell, is one that probably cannot be performed by software.

Oct 30, 2013 1:25 PM EDT

Reply

dstechDan

:)

Nov 1, 2013 8:53 AM EST

Reply

Whipster

There's a fairly obvious workaround/safety feature: dashboard engine kill switch that physically removes power from ignition and fuel circuits.... then you can relax a little on the firmware QC.

Oct 30, 2013 11:40 AM EDT

Reply

sh10453

@Whipster,
This maybe fine if the driver is an experienced automotive engineer or technician who is doing testing where the "kill switch" is always on their mind, and probably their finger is on that switch most of the time.
For a grandmother/mother/father/wife/husband, etc., who is starting the vehicle in their garage, and immediately encountering the UA, it would be far too late to even remember that there is a kill switch!
Even the experienced driver, who has his finger on the kill switch may not be able to push it in time.
This over-simplistic idea is OK in a control room that has an operator monitoring a process, such as an engine test cell.
Remember that reaction time, and time to act may take a couple of seconds or more, and that's a very long time when faced with a sudden acceleration, especially in closed spaces (garages, parking lots, etc.).
But I must agree that a kill switch would be a good idea to implement on a 1903, or maybe even on a 1920 vehicle!!!

Oct 31, 2013 11:09 AM EDT

Reply

LarryOsolkowski

Hmm, every motorcycle I've ever owned (several) has had a kill switch. Must mean something, but I'm not sure what.

Nov 1, 2013 3:46 PM EST

Reply

PowerStation

All software has bugs, ECC only protects RAM from flipping bits, not the CPU AlU from flipping bits, for instance. And it is impossible to predict the resutls of flipping one random bit, at a specific time, at a specific state.

The way you protect from problems like this is you add redundency into the system, you assume your software is going to nuts, so in this case the driver must be able to shut down the car, in an obvious way. Which apparently was not the case.

Oct 30, 2013 11:18 AM EDT

Reply

gemtech

In my opinion, the most basic flaw is that Toyota ran before they learned how to walk with "new technology". Too many any of the safety systems are running through the electronic control system: accelerator pedal, shift by wire, and pushbutton start/stop. Couple that with not informing owners that the start/stop button has to be held in for at least 3 seconds before the engine can be shut down (a major, major flaw that is being addressed by at least the US auto industry). I've looked through my 2007 Avalon owners manual and there is no mention of that that I could find. Their reasoning of not shutting it down for 3 seconds is flawed (again, in my opinion): without the engine running there will be no power steering assist (at 120mph you do not need power assist), without the engine running you will not have power brakes (at full throttle the engine is not making vacuum for the power brakes).

Oct 30, 2013 10:46 AM EDT

Reply

Michael Rathbun

@MrPWM: none of these functions stand on their own. Determining the TPS (Throttle Position Sensor) reading will involve an A-to-D converter start and finish (done in an interrupt service routine, if it is being done properly), followed by some checks for noise, followed by a normalization transform, followed by a reasonability check (hey, the circuit may have failed or the TPS may have gone west in the last 4.6 milliseconds) followed by a first derivative calculation to determine whether the throttle is opening or closing, and if so at what rate, and if so whether this is a throttle transient event that requires additional fuel to be added or subtracted to the fuel injection events currently scheduled in the Carousel.

If you can do that in 100 lines or less, my hat is well and truly off to you.

Oct 29, 2013 11:23 PM EDT

Reply

Michael Rathbun

And hey, I'll let you do it in C rather than 8065 Assembler.

Oct 29, 2013 11:45 PM EDT

Reply

MrPWM

Wow. Thanks for the explanation. All that for a "simple" throttle position sensor. Truly a design where nothing can go wro....go wro......go wro......

Oct 30, 2013 11:04 AM EDT

Reply

Michael Rathbun

Fascinating. I spent 8 years doing engine/powertrain control software for a large US and Australian motor manufacturer, and created a bunch of static and dynamic code analysis tools. At the top of my mind always was "My software can kill people."

Apparently that was worth considering, inasmuch as, TTBOMKAB, my software has not killed anybody.

Oct 29, 2013 11:08 PM EDT

Reply

Djbrettbsu

Anyone ever read about the Therac-25? Definitely not the type of device you want malfunctioning.

Oct 29, 2013 11:05 PM EDT

Reply

MrPWM

I'm analog, not a programmer but, for programming homework during college I wrote a lot of code for simple functions. All of it was usually less than 100 lines of code and that was all that was needed.
So digital guys, please tell me why are there "tens of thousands of lines" of code for the simple position of a gas peddle?

Oct 29, 2013 7:44 PM EDT

Reply

PowerStation

Ok, but first you tell me why you need hundreds of circuit board parts to make a ECM that controls one gas peddle (as shown in the picture above)? I have bought a simple eval board with 5 parts that could do it.

Oct 30, 2013 11:25 AM EDT

Reply

skyhooktek

safety, safety, safety

Oct 31, 2013 8:58 PM EDT

Reply

Work to Ride comma Ride to Work

Poorly designed firmware caused unintended operation, lack of driver training made it fatal.

Oct 29, 2013 3:25 PM EDT

Reply

kurtrad

I worked for GM Hughes Electronics in the 1980's. A group of us at Hughes Fullerton were tasked to investigate unwanted acceleration in some of their vehicles. The device was controlled by a small IC designed by AC/Delco (called acorn) that was state machine based and contained no software.

We analyzed and simulated the structural netlist defining the ASIC and could not find any bugs in the state machine logic. However we did find that if the power on ramp was too slow, then the flip-flops in the ASIC could power up in random states. There were a few hundred flip-flops in the design and were a handfull of states that they could power up in that could cause a the ASIC to command full throttle. The liklihood of this happening was nearly infintessimal, but yes there was a probability that it could occur. At the cruise controller system module level there were two traces on the board that were adjacent when shorted could cause the throttle actuator to command full throttle regardless of the brake pedal sensor. The main concern was if some metal shavings fell into the module this could cause unwanted acceleration.

My point of all of this is that systems such as these had problems even without the complexity of software and 'drive by wire' controls. Shortcuts caused by schedule pressure to deliver a product before it had thorough analysis for the consequences of failure mechanisms is always a tradeoff with delivery schedules for engineering projects.

Oct 29, 2013 3:06 PM EDT

Reply

mmerhar

So, a thought experiment. We have all heard the tales about Toyota's "zero defect" Andon manufacturing lines. If a tech should spot, for example, a batch of out-of-spec brake springs, the entire line is stopped until the source of the problem is found and corrected.

What do you think would have happened if a coder attempted to halt the line because they'd discovered a fatal stack overflow in ECM production code?

Until firmware (and the people and processes that produce it) are given the tools, respect, and authority available to those on the mechanical side of the shop, there will still be situations like this one. Just as there's a need for "certified engineers" signing off on safety-critical civil and architectural designs, there's probably a need for the same thing in the software space.

Oct 29, 2013 2:17 PM EDT

Reply

Curie_US

WOW! .... Glad I'm on my 3rd CAMRY in the past decade or so, each one with over 100K miles & NO sudden acceleration. My only regret is that they have all been auto transmission models. Too bad, too, since my left leg could use some exercise w/ the clutch pedal.

"....several materials handling systems that implemented LOTO functions in software ..." What is LOTO?

Oct 29, 2013 2:13 PM EDT

Reply

N2IXK

LOTO = "Lock Out, Tag Out". Procedures for securing energy sources to a device or system, typically for maintenance or setup purposes. Should always be done with PHYSICAL disconnect switches, valves, etc., and not trusted to software.

http://en.wikipedia.org/wiki/Lockout-tagout

Oct 29, 2013 5:43 PM EDT

Reply

atemp

As in politics and public-safety laws, somebody had to die before corrective measures were taken. There's a good reason why fly-by-wire systems have multiple-redundancy control and nav systems with forcibly-dissimilar platform architectures and firewalled software teams.

Hey Toyota, take a page from Boeing and Airbus, duh.

Oct 29, 2013 1:50 PM EDT

Reply

solutionseeker

NHTSA, with much sub rosa help from Toyota and Exponent, sabotaged NASA's efforts to identify the unintended acceleration problem in Toyota's vehicles. NHTSA officials did everything imaginable to impede their space agency counterparts, including lengthy initial delays, supplying bankers' boxes full of unlabeled components and grossly understated numbers of customer complaints and warranty claims, followed by a sudden, premature order that the investigation was "over." Secretary LaHood rushed to the microphones with the lie that "the verdict is in" and no electronic defects could be found by NASA. NASA was criminally betrayed by the people at NHTSA, and more people have died at the hands of Toyota as a result.

Oct 29, 2013 3:29 PM EDT

Reply

bcarso

This is well and truly an outrage, and I don't doubt for a moment that your account is accurate.

Now we are about to see the meltdown of what was once a great automobile company. So many similar things are coming to light now --- I need not enumerate them. The dimly-lit and the low-information set may only distantly acknowledge what is unfolding, but I think it will still roar ahead. Much as we enjoy disparaging the legal profession, they do have their efficacies, and I am delighted to see them turned loose. I had long suspected that Toyota had been in gross denial on this, and this is now confirmed to my satisfaction. And are they working now on self-driving cars, as my friend queried?

Oct 29, 2013 10:24 PM EDT

Reply

David Sherman

I appreciate the best technical article yet on the Toyota throttle problem, and I understand that drivers very reasonably expect automobile engines to create an amount of power that's monotonically related to the "gas pedal" position. However, I've never understood why even a totally stuck throttle, be it for mechanical or electronic reasons, should cause a wreck. On a typical car, the brakes can stop a car in about 1/10th the distance that the engine can accelerate it in. Looking at it another way, the brakes can absorb 10X the energy (for a short time period) that the engine can produce. It should always be possible to slam on the brakes and stall the engine, even if the throttle is stuck wide open. Alternatively, putting the transmission in neutral would also work. All modern engines are electronically governed so they won't over-rev, and even with an un-governed engine, it's better to throw a rod than to wreck the vehicle.

At least in my world, every wreck caused by a "runaway" engine is really caused by operator error.

Oct 29, 2013 1:36 PM EDT

Reply

Michael Dunn

A good reason to drive a standard—one is very used to the "neutral" position!

People have also mentioned turning off the ignition causing steering lock, but in many cases that would be acceptable. Besides, isn't there usually an intermediate key position that won't cause lock?

Do check out the video in the Resource section to see why brakes might not always work.

Oct 29, 2013 1:48 PM EDT

Reply

Thothsartre

If the car is a runaway, just turn it off! Every car I have encountered has an intermediate key position where the engine is off but the steering wheel is unlocked. I may or may not have driven many miles this way. You'll lose power brakes but typically not so soon that it would be a factor in an emergency slow-down, and there's always the emergency brake, too.

Oct 29, 2013 3:55 PM EDT

Reply

SPLatMan

Modern cars have keyless entry with a push button to stop/start the engine. That button connects to the ECM that just went ape.

It should be mandatory to have a killswitch wired directly to the ignition system, like an Estop on machinery.

Oct 29, 2013 4:20 PM EDT

Reply

Robert.Oppenheimer

Hmmm... Estop in the fuel pump power circuit is a quick fix. Unless there is a big accumulator, engine should die fairly quickly when fuel pump is shut down. Minimally invasive to other systems.

Oct 31, 2013 1:21 PM EDT

Robert.Oppenheimer

Even with an automatic, do a disaster drill on occasion. It's simplistic but on the open highway, floor the accelerator pedal to simulate a run-away and then turn off the key - or press and hold the Run/Stop button. Know how to regain control of a bad situation in an emergency. Next on the list- how to recover if your ABS/Stability control decides to lock up one wheel...

Oct 31, 2013 1:12 PM EDT

Reply

memsman

Braking a car with a stuck fully-open throttle without putting the transmission in neutral is very hard or impossible at almost any speed (think of momentum). The brake fade in roadgoing cars is just too horrible. The braking power drops off like a rock very quickly. My guess is at a speed of ~20mph or higher this is a runaway situation. The only way is to disengage or kill the engine.

Oct 29, 2013 1:54 PM EDT

Reply

benmlee2

It has been proven over and over many times even in the most powerful car full throttle, brakes can over ride engine.
Think about it, if you go up a hill at full throttle and reach X speed. Now come down that same hill at X speed. The energy would be

equivalent E=1/2MV^2. The brakes has to be able to stop the car. Right? That is how the car is designed. What you are saying is if you go up a hill at full throttle, car would not be able to stop coming down the same hill. That is simply not true. All brakes can over come the engine.

Oct 29, 2013 2:46 PM EDT

Reply

solutionseeker

At open throttle, the vacuum to boost the brakes is not being produced. One or two successive brake applications will deplete vacuum and, depending upon vehicle velocity, the driver may be called upon to apply in excess of 150 lbs. of brake pedal force, which is beyond the capability of most drivers. NHTSA's testing of a Lexus in 2007 demonstrated this fact.

Oct 29, 2013 3:18 PM EDT

Reply

Work to Ride comma Ride to Work

150 pounds? Really? Where did that piece of information come from?

Oct 29, 2013 3:40 PM EDT

SteveP67

That's why in this situation you stand on the brake pedal and hold it - you don't release because you will lose the vacuum assist.

Oct 29, 2013 8:26 PM EDT

oldcarguy

A few comments -- 150# isn't that much -- like standing on one leg (and doing a shallow squat) for someone that weighs 150#. It could be hard for a very small or feeble person, and of course would be completely unexpected by anyone used to a much lighter brake pedal force. Is there a reference for the NHTSA test?

If the brakes are applied hard *and* held on, they should always be able to overpower the engine. But--in some situations untrained (timid) drivers are likely to start with a light brake application which heats up the brakes. Once the brakes are hot (really hot) they may "fade" (boil the brake fluid and/or change the friction material properties) and no longer be able to overpower the engine.

A personal story -- thirty years ago, engines didn't warm up very quickly in winter (sub freezing temps). If I was cold in a rented car, I would often leave my motel in the morning and drive slowly with both brakes and throttle applied at the same time. After a little practice it became easy to control speed (perhaps 10-20 mph) with full throttle applied. This warmed the engine (and the cabin) up very quickly.

Sometimes the required brake pedal force would increase, as the vacuum booster lost power, but never enough to be a problem. Never faded the brakes, at that low speed the actual braking energy was low compared to the ultimate brake capability.

I kept some of those rental cars for a week, never had any follow on problems over that time period. Have done the same thing with my personal cars on very cold mornings with no lasting effects that I can detect--my last car was 20 years old when it finally rusted out. Brake life was normal.

Oct 29, 2013 8:45 PM EDT

bfw00001

All this speculation about braking ability is fun, but Car & Driver decided to do empirical testing. They found that holding the throttle wide open increased the Camry's 70-0mph stopping distance by only 12 feet. They tried several other models, and the only car they found where wide open throttle significantly increased stopping distance was a 500+ hp Roush modified Mustang.

Remember, what's recounted here is the testimony of a PAID expert witness. Look at the victim demographics - how do the software gremlins know the drivers' age, since 88% of SA incidents are with drivers 65 or older.

This has all happened before, to Audi, back when throttles were controlled by cables attached to the gas pedal. It was proven then that the engines couldn't overcome the brakes, but that didn't stop a few paid expert witnesses from convincing a few juries to award damages.

Oct 30, 2013 1:44 PM EDT

scrap

I have seen that proven with the vehicle at a stop. Add momentum and especially the rotational mass of the wheel/tire assemblies, which is much more energy to absorb per weight than the sprung weight of the rest of the car, and the reality is that a speeding car at WOT often can't be stopped with brakes. Brake fade and overheating will make the brakes even less effective until they become virtually worthless in an extreme case. The engine needs to be powered down somehow to bring a runaway vehicle to a halt. This may not be true for a wimpy powertrain attached to a light vehicle with massive brakes, but the majority of modern vehicles are built to proportion and suffer from the reality described above.

Oct 31, 2013 7:36 PM EDT

Reply

benmlee2

This is an experiment most high school kids has already done. Go full throttle and step on brakes. Your car will stop in no time. Really no need to debate. You can stop the car.
Retired officer in San Diego made a series of critical mistakes. Did not step on the brake hard enough, did not shift to neutral, did no drive off into the grass. There were plenty of tall shrubs to drive off into and stop safely. In another word, he had completely panicked. That leads me to think if he actually had his foot on the gas the entire time given his horrible panicked reaction.
Majority of older people that caused UA stepped on the wrong pedal and kept their foot there until they crashed.
I once stepped on the gas thinking it was the brake. (My uncle's car had a different pedal set-up for medical reason) The unexpected result was like a jolt of lightning thru your logic. It took every fiber in my body to lift the foot off. People got to realize the most un-safe, error prone, untested computer in the world is the one between your ears. That is what cause majority of accidents. That is why is called an accident. Fix that fisrt, and stop blaming it on some ECC chip.

Nov 1, 2013 11:12 AM EST

SteveP67

One of the early fatal cases involved a California Highway Patrol officer and his family in a rented Toyota. One would think he was a trained driver but he may have been a pencil pusher. Regardless, he wrecked at an estimated 120 mph killing himself, his wife, and two daughters. His wife had time to call 911 and scream for help while they accelerated to 120 mph. They couldn't shut down the engine. Why he didn't/couldn't put it in neutral was never addressed. Apparently Toyota's push-button ignition requires a 3 second press-and-hold to shut down the engine if the car is in gear (or perhaps moving - not sure which). This is understandable as you don't want to inadvertently shut off the engine if you accidentally hit the button. He was unaware of that since it was a rental he was unfamiliar with. He was repeatedly pressing the button in a panic. Other cars, such as Infinity, recognize a series of rapid pushes and don't require a long press-and-hold. Another bad software decision on Toyota's part.

As far as I know there are no cars that won't allow placing the transmission in neutral, so that would seem to always be a solution.

Oct 29, 2013 8:39 PM EDT

Reply

scrap

Your words here are very true and fair-handed. History shows that the off-duty officer, Mark Saylor, did not make the correct judgment while the vehicle was still at a controllable speed. His ES350 simply required pushing the shift lever forward one detent position into neutral. There is no gate or other input requirement he would have needed which could have impeded his already panicked state. In fact on that vehicle you can slam the shifter forward and it will stop in neutral (without the pushbutton being depressed), eliminating an accidental shift into reverse or park. It takes about 0.5 lb of force to overcome the detent. He could have done it with one pinky finger.

Agreed, Toyota should have made the push button of the Smart Key system more of an effective kill switch. They could also have employed an incremental throttle deration during concurrent throttle/brake request.

OTOH a driver should be fully familiar with a vehicle before taking it on the road where he or others can be harmed. As a supposedly trained driver per profession he certainly should have had the wherewithal to put the shift lever in neutral but he didn't. Anyone getting behind the wheel of a car should know this. Think of the time he had to react in a calm manner before the vehicle sped to dangerous speeds. That time is not a part of the 911 call. Imagine how long there was for him to recognize a WOT situation, relay that to the back seat passenger, and for the passenger to call 911 AND get a response.

Oct 31, 2013 7:30 PM EDT

Reply

scrap

Ran out of room...

People need to be ultimately responsible for their actions. Mark Saylor is the responsible party for not saving that carload of people's lives. If you think about culpability, the dealership employee that insisted on the all weather mat that caused the pedal entrapment is responsible for the situation, but the cause and effect of this tragedy are so far removed from each other, how can you place a verdict of homicide in any degree on such an innocent act? Sorry, I'm just venting some relatively random thoughts and chose your excellent post to jump in.

Oct 31, 2013 7:31 PM EDT

Reply

JHHayesII

Back in the late '60s when I was 14, a kid driving his car down a subdivision road smacked into a parked car because his accelerator pedal stuck. He had cranked around a corner and floored it. When he realized the pedal stuck, he reached down to pull the pedal up off the floor and lost control!

What do you do in an "Oh s$%t" moment? Your adrenaline starts to flow and you shift into panic mode! Now your thinking is not the best as you are on the verge of panic. When our physiology does this to us, well we all know about "fight or flight" but how many of us know about "play possum" or some derivative? Yeah you may just FREEZE! In any event your problem solving skills are compromised!

The preventative for poor panic responses during emergency situations IS practice.

When I was a kid taking drivers training we talked about brake failure. I took the concept to heart and contemplated what I would do if the brakes failed in the car I was driving. I was learning on the family car. A 1965 Plynmouth station wagon with a pedal emergency brake. I'd sit in the car and pretend that the brakes had just failed - so I'd jam on the floor emergency brake. I was 15.5 years old when I practiced this. So at 18 I'm driving down the road and the brakes failed, but I got the car stopped without running the red light. I drove home and told my mom what had happened so they'd know to get the thing looked at. Well, they failed on her too and she went right through the intersection. There was a problem with the brakes that occurred after some maintenance. The garage screwed up somehow and it set the brakes up for intermittent, unpredictable failure. It took four visits to the shop to fix that.

A stuck accelerator may leave the victim with more time to deal with the problem than a stuck brake, or possibly not. But you are in the worst possible situation to explore solutions with a limited amount of time.

I've not thought about stick accelerators. It's time to experiment and develop a plan!

Nov 1, 2013 11:39 AM EST

Reply

stratus46

I'm on my second Hybrid, a Prius and a Fusion. While there is a position on the shift lever called 'neutral', there is no actual 'neutral' in the 'transmission. Neutral in a Toyota / Ford hybrid is a state of the computer. The transmission has one planetary gear set with the engine driving the sun gear, a big electric motor/generator driving the ring gear and the output shaft on the planet 'carrier'. No reverse gearing or clutch of any sort. In other words, if the computer goes stupid, there is no 'neutral'.

Nov 2, 2013 4:18 AM EST

Reply

Michael Dunn

A reader just sent in three very interesting links:

http://www.latimes.com/news/opinion/opinionla/la-oew-cummings12-2010mar12,0,2595172.story

http://www.japanfocus.org/-David-McNeill/3993

http://www.carprogrammer.com/z28/pcm/FAQ/Delphi_Drive_by_wire_2000-01-0556.pdf

Oct 29, 2013 12:16 PM EDT

Reply

dvk

Nice article and great resolutions to follow.

Oct 29, 2013 5:39 AM EDT

Reply

The Measurement Blues

"it's impossible to test all potential hardware- and software-induced failure scenarios."
Because it's impossible to account for every failure scenario, lawyers can always find fault with such sayings as "is it possible..." The answer is always "yes," no matter how improbable.

Oct 28, 2013 11:29 PM EDT

Reply

FREng

Computer scientists have known for at least 40 years that "testing can only show the presence of errors, never their absence". Software developers should know the relevant computer science - especially if they consider themselves engineers.

If you want to know the properties of some software, for ALL inputs, you need to analyse it mathematically. Software engineers have known how to do this since Alan Turing's time, and several companies have shown that it is both practical and cost-effective to use mathematically formal development methods for software. For an example, type "tokeneer" and "praxis" into your favourite search engine.

Oct 29, 2013 11:43 AM EDT

Reply

Michael Dunn

I remember a period (1990s?) when "mathematically/formally correct" methods seemed to be in the air (and the journals). Haven't heard much about them since. What happened?

Oct 29, 2013 11:52 AM EDT

Reply

Nichts

In the Ada world they are trying to continue this with Spark see **http://en.wikipedia.org/wiki/SPARK_(programming_language)**

Oct 29, 2013 10:04 PM EDT

Reply

Dataflow

ACL2 is a programming language that has made massive strides in provability. It has been used to prove out the correctness of the floating point stack for CPUs and prove that one version of Linux had zero buffer overflows. **http://en.wikipedia.org/wiki/ACL2**

Oct 30, 2013 11:08 AM EDT

Reply

Stuart Matthews

What happened is that we carried on developing high integrity systems using SPARK. The SPARK programming language and its associated tool set and development methods are specifically designed for engineering safety-critical and security-critical systems. They do this in the way that you suggest – by having a firm mathematical underpinning that allows the elimination of common programming errors way before they ever enter service. See **http://intelligent-systems.altran.com/technologies/software-engineering/spark.html** for a list of the kind of systems on which we and our customers have successfully used SPARK to produce high-assurance systems.
This formal approach to software engineering is still very much alive and continuously evolving. We are actively working on the next generation of the language and toolset, SPARK 2014, as described here: **http://www.spark-2014.org/**

Oct 30, 2013 1:49 PM EDT

Reply

The Measurement Blues

I guess I'll keep my 1999 Camry for another 100,000 miles, then get a bicycle.

Oct 28, 2013 11:27 PM EDT

Reply

Michael Dunn

I don't think my 1998 has another 160Mm in it. Hello bi/tri/quadracycle.

Oct 29, 2013 9:58 AM EDT

Reply

The Measurement Blues

"it's impossible to test all potential hardware- and

software-induced failure scenarios."
Thus, even with best design practices, intense design reviews, and stringent testing, errors can still occur.

Oct 28, 2013 11:25 PM EDT

Reply

sixtysixscrews

'Thus, even with best design practices, intense design reviews, and stringent testing, errors can still occur.'

From what I read in this article, best practices and stringent testing weren't part of the process - they weren't even in the room:

-ECC memory wasn't used (although Toyota claimed it was)
-stack overflow conditions were not exhaustively researched
-single bit errors could result in loss of throttle control
-the CPU vendor-supplied RTOS version [OSEK] was not certified compliant (with what?)

When you are controlling a 1 ton+ piece of equipment that can move in excess of 120 km/hr you better be sure the fly-by-wire software works right. Lockheed learned this in spades with the F-16 program in the '70s - ask the widows and children of the pilots killed by bad software. I don't know if they got anything but a 'thanks for your sacrifice' from the government, but civilians shouldn't be assuming the role of test dummies for what amounts to an integration vendor's work.

I used to keep a file of software bugs that killed people but stopped updating it in the late '80s. By then it included a buggy user interface on a medical irradiation machine (two deaths) and several materials handling systems that implemented LOTO functions in software (five deaths, mostly from massive trauma).

Note that NHTSA does not have any standards for software even though software has taken on most of the command-and-control functions that once were performed by hardware. This is a reflection of the weakness of the agency - and can be directly traced to the ideology of politicians who would further weaken its powers.

On the other hand, the FAA has standards for avionics software - and I don't know of a civilian aircraft death that can be traced to software.

I used to have a bumper sticker that read 'Speed kills.' I think I need a new one: 'Software kills.'

wb

Oct 29, 2013 1:06 AM EDT

Reply

The Measurement Blues

"Software kills" brings a whole new meaning to the word "crash."

Oct 29, 2013 9:33 AM EDT

Reply

PowerStation

For avionics, it's not the FAA software standards but quad redundancy with different architectures is what makes it so reliable. You have four independent computers, all using different processors, different code, different software teams, different A/D's, different D/A's etc, all comparing their results together. All have errors, yes, but none of them have the SAME error. Three will outvote the forth when one of them fails to work properly. It is the only method that works.

Oct 30, 2013 12:01 PM EDT

Reply

Michael Dunn

I've wondered about redundant systems. The "voting machine" is still a single point of failure, albeit generally a much simpler one than the "voters", and easier to make reliable.

Oct 30, 2013 12:48 PM EDT

Reply

PowerStation

There is no literal voter, each motor has four seperate windings, three outvote the forth, by sheer current driving into the motor.

Oct 30, 2013 4:56 PM EDT

Jon D Hagar

Actually, read on the so called "N-version" programming problem. The study has shown different software does have the SAME error in some percentage of the bug cases. This was well studied many years back. Yes FAA and NASA still like it, but multiple version don't solve all reliability problems. Bottomline: We will put software everywhere, but one must have many approaches to improving system quality, e.g. specification, design, coding, and testing. Sorry I was late to this party. Very interesting piece.

Nov 18, 2013 2:45 PM EST

Reply

bfw00001

From what you read, a PAID expert witness CLAIMED they weren't part of the process. The article does not mention cross-examination about things such as his visibility into their internal process.

Oct 30, 2013 2:18 PM EDT

Reply

FREng

I repeat: Computer scientists have known for at least 40 years that "testing can only show the presence of errors, never their absence". Software developers should know the relevant computer science - especially if they consider themselves engineers.

If you want to know the properties of some software, for ALL inputs, you need to analyse it mathematically. Software engineers have known how to do this since Alan Turing's time, and several companies have shown that it is both practical and cost-effective to use mathematically formal development methods for software. For an example, type "tokeneer" and "praxis" into your favourite search engine.

Oct 29, 2013 11:43 AM EDT

Reply

asdfasdfsdfsdf

Well, to be fair, what Turing did was prove that you *couldn't* analyze software mathematically. :-P

Oct 29, 2013 5:23 PM EDT

Reply

pastorofmuppets

The proof was for arbitrary programs. There are techniques to deal with minimizing the input sample space, as well as what operations are to be performed, and still be able to reason about the behavior of certain programs.

Oct 29, 2013 6:01 PM EDT

Reply

Hodapp

What Turing proved is that a *program* could not, in the general case, determine whether or not an arbitrary program in a Turing-complete system would terminate. If you remove Turing-completeness (which is not actually necessary for a lot of software), if you're concerned not with termination but with some other invariants, or if you're talking about any sort of proof that you're not expecting an algorithm to write for you - mathematical analysis is still quite a viable method.

Oct 30, 2013 9:30 AM EDT

Reply

Dataflow

ACL2 is a programming language where you declare post conditions for functions. The output of the ACL2 compiler is not only the EXE but also a somewhat human-readable proof of correctness (or incorrectness if compile fails) for those post conditions and any calls that fail to meet the preconditions. Textbook here: http://www.cs.utexas.edu/users/moore/publications/acl2-books/car/

Oct 30, 2013 11:19 AM EDT

Hodapp

Very interesting. This reminds me a bit of languages like Agda, but Agda comes more from a Haskell background relying heavily on static typing. This is the first language of the sort I've heard of that derived from a Lisp.

Oct 31, 2013 12:10 PM EDT

Most Popular Most Commented

Toyota's killer firmware: Bad design and its consequences

Hybrid automotive use of ultracapacitors

Teardown: OBD-II Bluetooth adapter

Fundamentals of the automotive cabin climate control system

Cars run HTML5-based applications

Engineer shares how to build an electric vehicle from the ground up -- Part 1: Design choices

Teardown: Heads-up thermal imaging camera

Automobile sensors may usher in self-driving cars

Teardown reveals Chevy Volt's electronic secrets

Engineer shares how to build an electric vehicle from the ground up -- Part 1: Lead-acid vs Lithium-ion batteries

RELATED CONTENT

Accelerating toward $1 million prize

It's the car! No, it's the driver! Wait, it doesn't matter!

Toyota Announces Priority Registration Web Site for All-New Prius Plug-in Hybrid

Toyota Announces Second Annual Shareathon Program

Toyota Announces Marketing Campaign For The Reinvented 2012 Camry

All of the event info you need @ your fingertips
DesignCon 2013
Download the DesignCon App!

FEATURED RESOURCES

Subscribe to RSS: or

DESIGN CENTERS

Analog
Automotive
Components &Packaging
Consumer
DIY
IC Design
LEDs

Medical
PCB
Power Management
Sensors
Systems Design
Test & Measurement
Wireless/Networking

MORE EDN

Blogs
Design Ideas
Tech Papers
Courses
Webinars

EDN TV
Events
About Us

SUBSCRIBE TO NEWSLETTERS TODAY! DON'T MISS ANOTHER ISSUE OF EDN IN YOUR INBOX!

GLOBAL NETWORK **EE Times Asia** **EE Times China** **EE Times Europe** **EE Times India** **EE Times Japan** **EE Times Korea** **EE Times Taiwan** **EDN Asia** **EDN China** **EDN Japan** **TechOnline India** **ESC Brazil** **ESC India**

FEATURED UBM TECH SITES: **EE Times** | **EBN** | **EDN** | **DataSheets.com** | **Design News** | **Embedded** | **TechOnline** | **Planet Analog**

OUR MARKETS: **Business Technology** | **Electronics** | **Game & App Development**

Working With Us: Advertising Contacts | Event Calendar | Tech Marketing Solutions | Corporate Site | Contact Us / Feedback