Rui Gaspar, Benno Wiesner, Gunther Bauer

Virtualization allows the simulation of automotive ECUs on Windows PC in closed-

-

formed faster and cheaper *n this paper, we report the implementation of this idea for the case of B+W,s - speed transmission ./ -%P !he technical challen"e is0 %ow to port ECU tas$s and #asic software to Windows PC with reasona#le effort, so that $e) development tas$s can #e performed on a PC, without the need of access-in" real hardware such as vehicle protot)pes, test ri"s or %i& facilities 1ince different parties '2E+ and suppliers(3ointl) developed the !CU, the protection of intellectual propert) 'models and source code(re4uired special attention as well

! " # $ %

5utomotive control software is of often 3ointl) developed #) an 2E+ and suppliers as shown in /i" 6 !)picall), a team of function developers uses a model-#ased tool chain to develop a model of the ECU and to "enerate C code from that !he resultin" C code is then compiled for the tar"et processor, and the resultin" ECU is validated

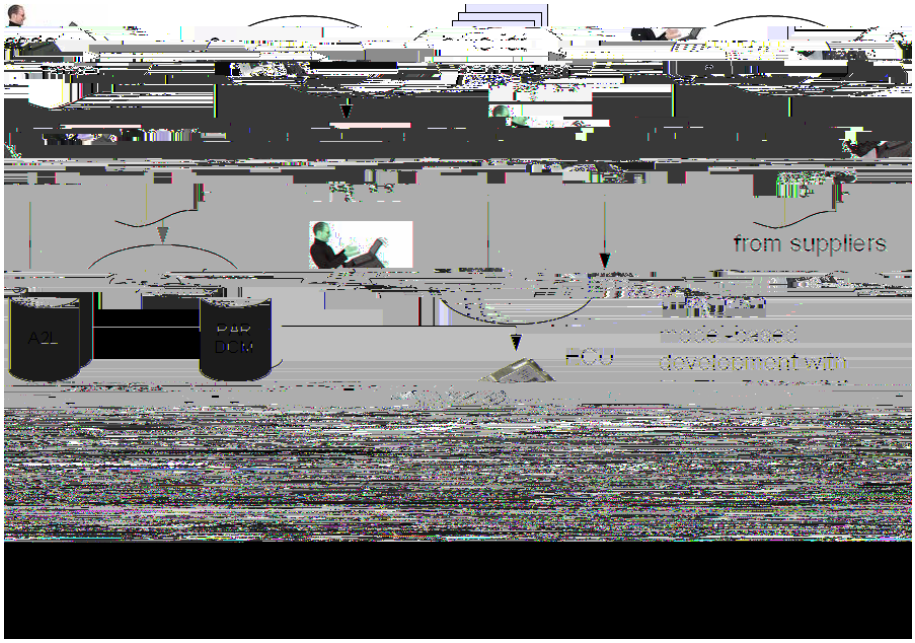

Fig. 1: Development process for automotive software

and tested usin" test ri"s, %i& s)stems, and road tests !he test and validation results are fed #ac$ to the developers, which closes the development c)cle !his process, althou"h standard in the automotive industr) toda), has two ma3or draw#ac$s0

a sin"le iteration ta$es da)s or wee$s0 feed#ac$ reaches developers late

the process depends on protot)pe vehicles and test ri"s !hese are t)picall) scarce and e:pensive resources durin" development !heir limited availa#ilit) causes additional dela)s durin" development

!his paper demonstrates how to improve the process !he $e) idea is to provide each development en"ineer with a virtual ECU. !his can #e simulated, cali#rated and measured on the developer,s laptop - either in closed-loop with a vehicle simulation model, or in real-time for rapid control protot)pin" !his wa), more development tas$s can #e performed faster and cheaper on the developer,s laptop 5s e:perience shows, this helps to shorten development c)cles and to reduce the critical dependenc) on scarce resources and real hardware

!here are two main options to set up a virtual ECU on PC0

Re-host the native binary co e using chip simulation. !he native ECU code '#inar)(is e:ecuted on PC #) emulatin" the instruction set of the ECU processor ;<= !his re4uires no access to the C code

Re-target the C co e Compile the C code of the ECU for e:ecution on Windows PC 2#viousl), this re4uires access to the C code to #uild a Windows e:ecuta#le or >&&

!he paper is structured as follows0 *n the ne:t section, we descri#e how we have virtualized the transmission controller of the ./ %P- automatic transmission, and list $e) differences #etween the real and the virtual !CU 1ection < presents some applications of this virtualization 1ection 9 concludes with a #rief outloo$ on future wor$

& '( )*

!he ./ -%P is an ei"ht-speed automatic transmission !he ei"ht "ears are implemented usin" four planetar) "ear-sets controlled #) five h)draulicall) operated #ra$e and clutch elements as shown in /i" 8

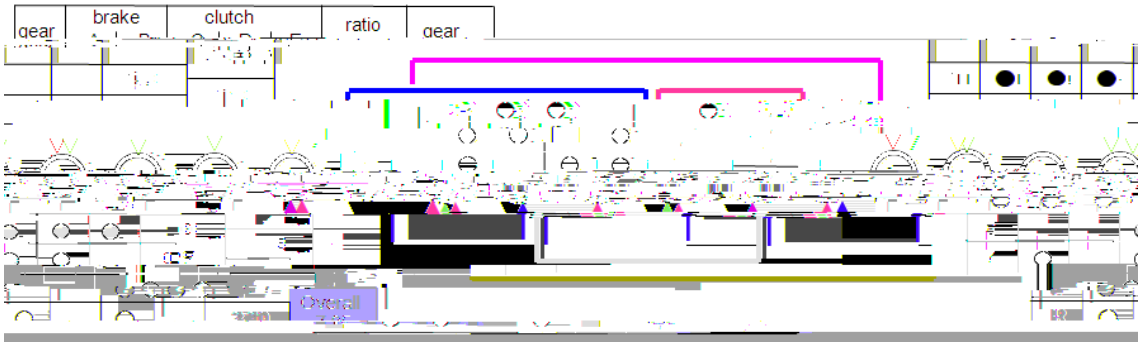

Fig. !: "ears an possible gear shifts of the #F $%& $-spee transmission

!he shift strate") of a transmissions is often developed #) an 2E+, while the re-mainin" control software comes from a supplier /or the ./ -%P transmission con-sidered here, a#out <7? of the shift strate") module is "enerated with !ar"et&in$ from a model developed with +5!&5B@1imulin$!he remainin" part of that module is hand-coded usin" C

We used the virtual ECU tool 1ilver ;8= to re-tar"et the entire control code as a Windows d)namic lin$ li#rar) 'dll(We could have used 1ilver,s chip simulator ;<= as well, #ut decided to use here the variant #ased on re-tar"etin" for two reasons@

Re-tar"eted code runs a#out 67 times faster on a PC than a correspondin" chip simulation !herefore, chip simulation is most useful in cases where re-tar"etin" the C code is not possi#le

!he control software for the -%P transmission was partl) developed usin" the virtual ECU tool 1oftCar ;6= /or that reason, a re-tar"eted version of the con-trol software was alread) availa#le as Windows li#raries ' li# files created with +1 Visual 1tudio(

1ilver provides a framewor$ for re-tar"etin" control code to Windows !his frame-wor$, called 1ilver Basis 1oftware '1B1(, uses the same source files that are avail-a#le durin" the normal ECU development, #ut bypasses the ori"inal #asic software of the ECU with services supplied #) 1ilver@

!he R!21 is replaced #) a simple e:ecution loop in 1ilver !his runs tas$s either initiall), periodicall) with defined offsets, or at certain events 'interrupts(

!he >BC files descri#in" the C5A communication of the ECU are used to emulate C5A processin"

!he 515P8@58& file descri#in" all tuna#le and all measura#le varia#les of the ECU is used to #)pass analo"ue input and output processin" of the ECU /or e:ample, instead of simulatin" the low-level processin" of a certain pulse-width modulated 'PW+(si"nal that encodes the tar"et current for a ma"netic valve, we directl) use the correspondin" hi"h-level current varia#le !his mi"ht #e a 6B-#it unsi"ned inte"er, which - after application of an associated scalin" rule - represents the tar"et current in 5mpere !he scalin" rule is part of the 58& description of the inte"er varia#le 1ilver $nows how to appl) directl) and how to invert the rule, and uses this to automaticall) convert the raw inte"er values to ph)sicall) meanin"ful values durin" simulation 'and vice-versa(!his wa), low-level processin" of the #asic software can #e easil) #)passed *n our e:ample, the tar"et current in 5mpere is directl) fed into the simulation model of the ma"netic valve, which is part of the vehicle model descri#ed in section 8< 5 similar mechanism is used to #)pass the low-level sta"es '5> conver-sion, si"nal filters(used for sensor value ac4uisition

We used the a#ove framewor$ to re-tar"et the 5G1 module of the !CU, #ased on the C code, #oth hand-coded and "enerated !his too$ us a#out three person da)s !o validate the result in 1ilver, the re-tar"eted 5G1 module was then simulated us-in" measured inputs !he PC simulation commanded e:actl) the same "ear shifts as those measured in the real vehicle Encoura"ed #) this 4uic$ success, we re-tar-"eted then the entire !CU, #ased on the Windows li#rar) received from ./

& & + , - % .

a#le(plant models, or couplin" with a source-level de#u""er '#rea$points,

!he PC simulation of the -%P transmission s)stem can #e used as follows0

/pen-loop analysis of measurements on &C0 Use measurements 'e " a +>/ or >5! file(ta$en on the road or on test ri" to drive the virtual ECU on PC !his wa) it #ecomes possi#le to loo$ at all !CU varia#les in detail 'lets sa), 677 777(!his "ives a fairl) complete picture of the ECU #ehaviour !his usa"e of a virtual ECU is the easiest to implement #ecause it does not re4uire an) vehicle model

Debugging on C source level0 5ttach the +1 Visual 1tudio de#u""er to the virtual ECU runnin" in 1ilver to de#u" pro#lems on C code level 2n a real ECU, a runtime e:ception li$e an inte"er ivision by 0ero or a memory access violation will t)picall) tri""er an ECU reset !hese $inds of pro#lems are difficult to catch and anal)se in real-time environments, e " on a %i& s)stem or on the road With the virtual ECU, there is no real-time constraint0 1imulation can #e stopped to inspect varia#les, call

1