**AUTOSAR  
lecture 3**

**Interrupts type in autosar**

|  |  |
| --- | --- |
| **CAT1** | **CAT2** |

CAT2 category 2

ال OS هوا الى مسؤل يشغل ال interrupt دا ... انه ي

Enable interrupt  
control priority for this interrupt  
make the context switching of this interrupt

طب ما كدا كدا ال compiler هيعمل ال context switching ؟

... هنا فى context switching مختلف هيعمله ال OS ذى

انه يعمل save لل state بتاعت ال task الى فاتت ... لان فى ال OS ممكن ال interrupt يشغل task اعلى priority

طب هيشغلها ازاى ؟ ... ال OS و هوا خارج من ال ISR بيشغل ال scheduler عشان يشوف انهى تاسك الى فى ال ready state و المفروض تروح لل running بسبب ال interrupt دا

CAT1 category 1

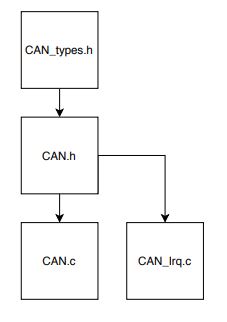
هنا ال OS مالوش علاقه بال interrupt دا ...

دا هيتعامل ذى اى interrupt هيقطع ال task الى شغاله و لما يخلص ال ISR هيرجع يكمل ال task الى كانت شغاله

فممكن يكون ال interrupt دا يشغل task اعلى priority بس ال OS مش هيروح لل task دى ... ليه ؟ لان ال ISR الى فى CAT 1 مشغلتش ال scheduler فال task دى هتتاخر

**ISR interrupt service routine**

1. مش انا الى بعملها call ال HW هوا الى بيعملها call فبالتالى بتاخد void و بت return void
2. فى ال autosar ال ISR بتاعت ال module بتتحط فى file لوحدها اسمه <MSN>\_Irq.c
3. وظيفته انه يدينى notification ان ال interrupt دا حصل بس
4. بالتالى ال file dependencies لل CAN module مثلا هتبقى كدا



**Call back function**

* ال call الى بيحصل من layer فوق ل layer تحت اسمه direct call
* بستعملها كانها notification ذى ان مثلا ال
* APP send to CanIf a frame
* CanIf send the frame to CanDriver
* CanDriver sends the Frame and notify the CanIf that the frame sent
* CanIf notify the APP that frame sent
* These notifications done by callback

ازاى ؟ ... بال pointer to function

ليه ؟ عشان مينفعش module فى lower layer يعتمد على module فى higher layer

**Error reporting**

فى نوعين من ال errors فى ال AUTOSAR

|  |  |
| --- | --- |
| Development error | Production error |

انواع ال errors الى ممكن تحصل و ال ECU شغاله

|  |  |
| --- | --- |
| Hardware error | ذى ان يبقى فى voltage error فى بطاريه العربيه  او memory حصلها damage |
| Software error | ذى انى بكتب فى ال eeprom و بعتلها address مش موجود عندها |
| System error | ذى ان يحصللى overflow فى ال RX buffer بتاع اى communication module  او LIN time out ان وقت ال frame جه و انا مبعتتش ال frame |

بالتالى هنا كل module بيعمل include لل DET module عشان يعرف يreport ال development error

**Det\_ReportError**

لما ال code يطلع production بعمل disable لكل ال modules DET

ال function بتاعت ال report error بتعمل ايه جواها ؟ .... على حسب الى انا عايزه ملهاش standard

ممكن احط جواها while 1 عشان توقفلى الكود فاعرف ان فى error و اخش اشوفه بال debugger

او ممكن اخليه يبعت ال errors دى ل file معين او سيرفر معين

لو ال channel Id اكبر من عدد ال pins بتاعت ال microcontroller

**void** **Dio\_WriteChannel**(Dio\_ChannelType ChannelId , Dio\_LevelType Level){

**#if** (DIO\_DEV\_ERROR\_DETECT == STD\_ON)

**if**(ChannelId >= DIO\_MAX\_NUM\_CHANNEL){

Det\_ReportError(DIO\_MODULE\_ID ,

DIO\_INSTANCE\_ID ,

DIO\_WRITE\_API\_ID ,

DIO\_E\_PARAM\_INVALID\_CHANNEL\_ID);

}

**#endif**

}

ال null pointer error

**void** **Dio\_init**(**const** Dio\_ConfigType\* ConfigPtr)

{

**#if** (DIO\_DEV\_ERROR\_DETECT == STD\_ON)

**if**(ConfigPtr == NULL){

Det\_ReportError(DIO\_MODULE\_ID ,

DIO\_INSTANCE\_ID ,

DIO\_INIT\_API\_ID ,

DIO\_E\_PARAM\_CONFIG);

}

**#endif**

}

**Development error**

دا نوع ال error الى بكتشفها و انا لسه بعمل design لل system

* ال errors دى بنعملها report لل DET module ... دا موجود فى ال service layer اختصار Development error tracer
* مين الى بيعمل reporting لل development errors ؟ ... كل ال basic software modules
* ال errors دى موجوده فى ال SWS requirements بتاعت كل module
* ال error reporting ممكن اعمله enabling , disable لكل module
* ازاى اعمل report error ؟ ... بعمل call لل function

**void** **Det\_ReportError**(

uint16 ModuleId,

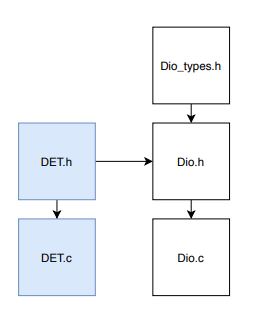
uint8 InstanceId,

uint8 ApiId,

uint8 ErrorId

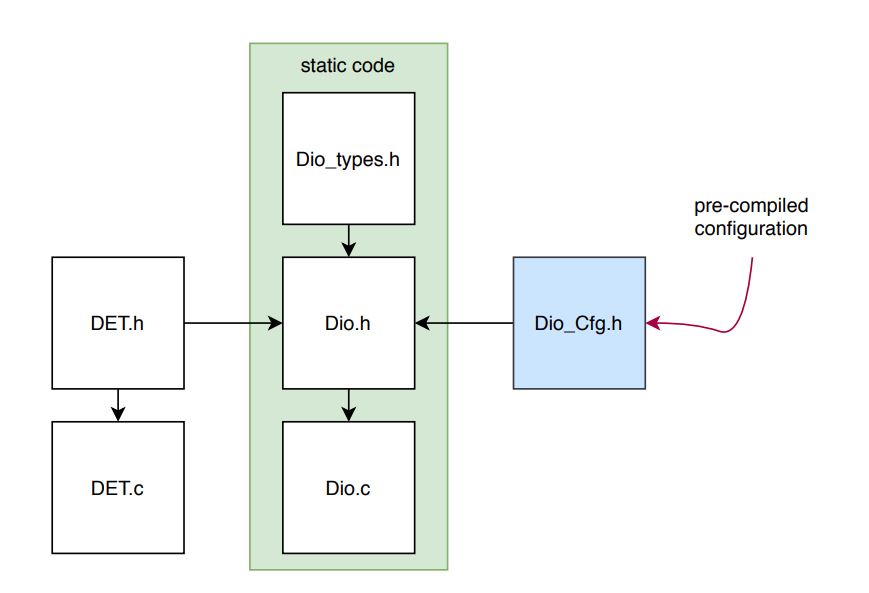
){}

|  |  |
| --- | --- |
| ModuleId | كل module ليه ID بيميزه |
| InstanceId | لو عندى CAN internal و CAN external فكدا عندى instance 0 , 1 عشان اعرف بالظبط انهى driver الى حصل فيه المشكله |
| ApiId | بيعرفنى ال function الى حصلت فيها المشكله |
| ErrorId | المشكله نفسها (بتبقى موجوده فى ال SWS ) |

****

**Static Configurations**

* دى ال configurations الثابته اثناء ال runtime ذى ارقام ال enable pins بتاعت ال external WDT او ال LCD
* بستخدم ال static configuration لو عايز اعمل enable او disable ل functionality معينه ذى ال DET module مثلا
* لو عايز اتحكم فى ال existence بتاعت function معينه فى ال module ذى ال Dio\_flipChannel()
* بتتعمل بال #define
* نوعها pre-compile configuration
* بنحطه فى ال <MSN>\_Cfg.h



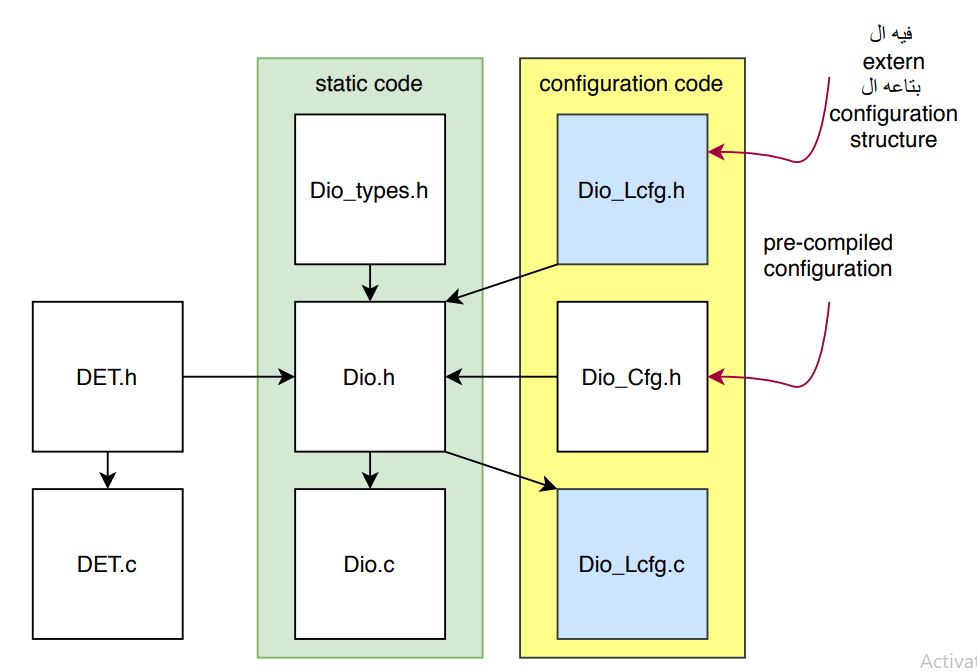
* خلى بالك :

هنا انا لو عدلت فى اى configuration هضطر اعمل build للكود كله تانى و دا عيب لانه بيزود ال compiling time

كمان لازم يبقى معايا ال source code عشان اعرف اعمل configuration و build تانى

**Link time configurations**

* هنا ال configurations ممكن تتغير فى ال run time ذى ال timer mode فهعملها فى شويه C variables او اجمعهم فى configuration structure



* كدا الى هيستعمل ال DIO مثلا يقدر يدى لل Dio\_init ال structure جاهزه بس مجتاج يكتب extern لل structure دى ... فيائما اكتبها فى ال Dio.h او اعمل فايل كمان فيه ال extern بتاع ال structures الى فى ال link time configuration file
* خلى بالك :

هنا انا لو عدلت فى اى configuration هضطر اعمل link بس لل Dio\_Lcfg.c مع باقى الكود فبالتالى قللت ال build time لانى عملت rebuild بس لل Dio\_Lcfg.c

كدا دا بيمكننى انى اقدر اعمل configure لكود C انا مش شايفه اصلا ... ممكن حد يبعتلى الكود بتاعه .o (object file ) و انا اغيرال configuration الى انا عايزها و اعمل link تانى بس مع الكود بتاعه

**Configurations**

الهدف من ال configurations انى اشغل ال module عشان اغير فى ال bahaviour منغير ما اغير فى كود ال C بتاع ال module

بنقسم الكود ل

|  |  |
| --- | --- |
| Static code |  |
| Configuration code | دا الى بيظبط ال module behavior على حسب الى انا عايزه |

ال configurations تلت انواع

|  |  |
| --- | --- |
| Pre-compile configurations | #defines |
| Link time configurations |  |
| Post build configuration |  |

**Production error**

ال error الى موجود و الحاجه شغاله يعنى و ال ECU شغاله

الى بيستقبل ال production errors هوا module اسمه DEM او diagnostic Event manager

مين الى ليه انه يبلغ عن production error ...؟ ال basic software layer modules الى ليهم انهم يبلغو ( مش كلهم ) و ال Application

ال error دا بيتخزن

الى بيبلغ عن error بيعمل call ل function اسمها

Std\_ReturnType **Dem\_ResetEventStatus**(

Dem\_EventIdType EventId)

{

فى الكود نفسه بنخزن ال ID دا فى flash memory لكل ECU

}

مين الى بيحط ال EventId ؟ ... انا .... لان كل واحد عايز يعمل check ل error على مزاجه

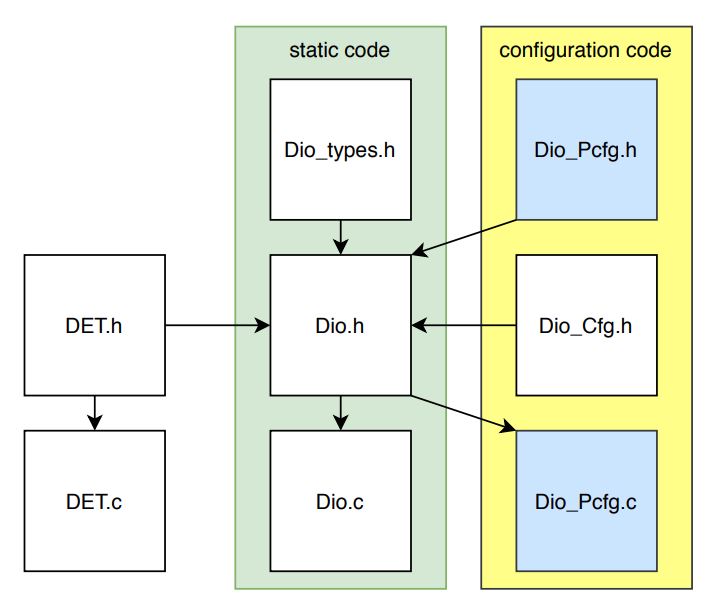
و ليه status سواء success او fail ... عشان اعرف حاله ال event دا

**Post build configurations**

* هنا انا لو غيرت member واحد فى ال configuration structure محتاج بس اعمل reallocation للقيمه الجديده فى مكانها فى ال memory منغير ما اعمل build تانى او linking تانى

و كدا كدا ال Dio\_init معاه عنوان ال structure فى ال memory

هنا ممكن اعمل re-flashing لجزء فى ال memory



* ليه محتاج اعمل كدا ؟ ... عشان اسهل ال testing انى اغير ال configurations و اflash و اجرب كل الاحتمالات بكل ال configuration

**Configurations is a generated code**

ال configuration بتتحط فى حاجه اسمها containers و كل container جواه container او parameters

1. Multiplicity

عدد ال configuration ... المفروض اعمل من ال container دا كام واحد

مثلا ... فى chapter 10 فى ال DIO module

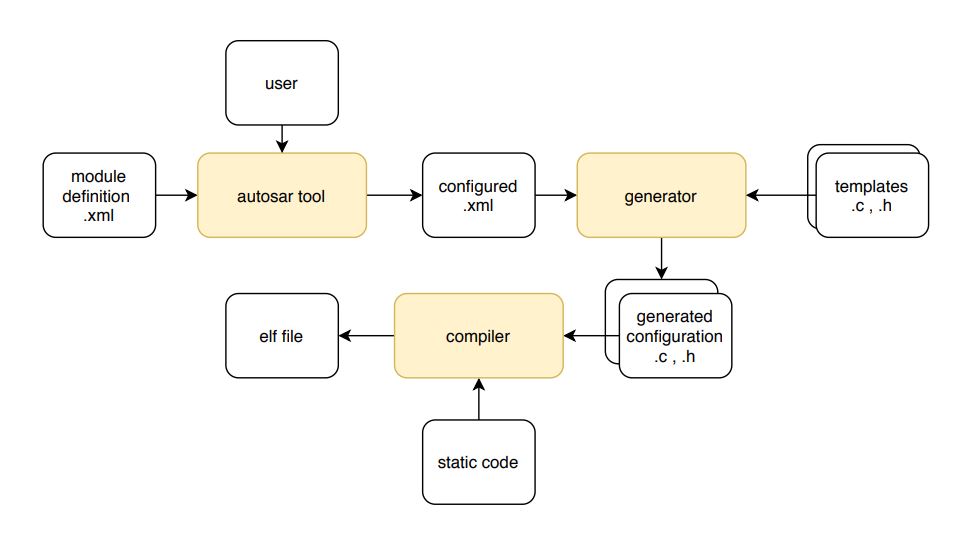
دا الجزء الخاص بانى اشغل او اطفى ال development error لل Dio module و دا جزء من ال container الى اسمه general

|  |  |  |  |
| --- | --- | --- | --- |
| ***SWS*** ***Item*** | **DIO142\_Conf :** | | |
| ***Name*** | DioDevErrorDetect {DIO\_DEV\_ERROR\_DETECT} | | |
| ***Description*** | Switches the Development Error Detection  and Notification ON or OFF | | |
| ***Multiplicity*** | 1 | | |
| ***Type*** | EcucBooleanParamDef | | |
| ***Default value*** | -- | | |
| ***ConfigurationClass*** | ***Pre-compile time*** |  | All Variants |
| ***Link time*** | -- |  |
| ***Post-build time*** | -- |  |
| ***Scope / Dependency*** | scope: Module | | |

ال multiplicity = 1 لانى لازم اعمل ال configuration دى مره واحده

ال type = Boolean عشان دى يائما on / off بالتالى فى ال autosar configuration tool هيبقى جمبها check box

ال configuration class فى بتحدد ال configuration دى هيا انهى نوع من ال configurations الى قلنا عليها ... هنا هيا precompile



1. ال autosar driver بيتسلم معاه ال module definition دا xml file عشان بس يقول لل tool تعرض ايه بحيث مثلا فى ال DIO هتعرض لل user انه ممكن بعمل container من نوع DioPort طب مين الى قال لل tool اساسا ان فى container من النوع دا فى ال DIO ؟ ... ال XML file دا
2. ال autosar tool بتسمح لل user انه يعمل containers جديده من الانواع الى موجوده فى ال XML file و تسمحله يعمللها ال configuration فى شكل GUI
3. يطلع من العمليه دى XML برضو بس فيه القيم الى انا حاططها ك user
4. ال XML دا يخش على ال generator و يخش معاه templates (دى عباره عن ال structures فى فايلات C , H بس منغير القيم بتاعتها ) مين الى هيحط القيم بتاعتها ال generator هيقرا من ال XML و يملا ال .c , .h بتوع ال templates
5. ال generator بعد كدا هيطلع كود C , h files بس مليانه بال configurations الى انا حطيتها تخش عالكمبايلر او ال linker و يطلع ال elf file