

AUTOSAR

lecture 1

Software requirements

هوا عبارة عن شرح او اقتراح لل software architecture الى هتتحقق ال functionality المطلوبة من ال software دا

طيب هيا ايه ال requirements دي ؟

1- Functional requirements

هيا requirements خاصة بال behavior او ال tasks المطلوبة من ال SW دا

a. Positive requirements (useful function)

ذی ان لازم ال function دی لو دخلتها ال input كذا هيطلع output كذا ... الى هوا الحاجه المطلوبة فعلا من ال function او ال software

b. Negative requirements

ان مثلا لو دخلته input غلط هي report error او مينفذش حاجه معينه

2- Nonfunctional requirements

a. Real time constrains

b. Resource constrains

ذی انی مضطر اشتغل على controller معين

3- Other requirements

ذی انی لازم مثلا اكتب بال autosar standards

Software design

ال software design بيتقسم ل :

1- Static design

ال static design هوا انی اقسام ال software بتاعی لاجزاء كل جزء مسؤل عن function معينه كذا ال system بقى مش complex و بقى اسهل فى ال testing , debugging بدل ما ابقى ب test ال system كله على بعض افكر :

- ال module testing هوا انی ب test كل module لوحده

... بتاع ال timer لوحده و ADC لوحده و هكذا

- ال integration testing انی اعمل test لكذا module مع

بعض عشان اشوف ال interfacing بينهم و بين بعض

- ال system testing ب test ال functionality بتاعت ال

system كله مع بعض

- فى test اسمه code review ... دا معناه ان حد بيص

عالكود بتاعى عشان يتأكد من حاجات مينفعش يتعملها test

ذی مثلا اسماء ال functions

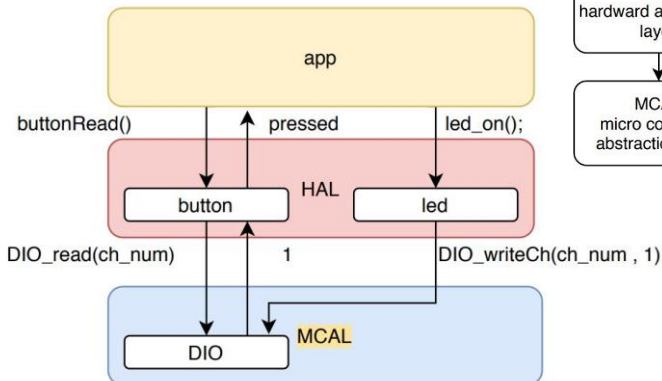
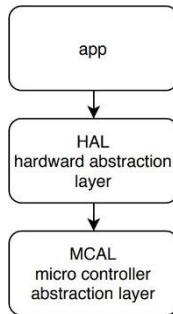
ال static software design بيبيد فى ال reusability انی اقدر

اجيب module كنت كاتبه قبل كذا و استخدمه تانى فى كل project

Layers

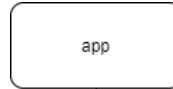
ال layers هيا جزء من تقسيمه ال code فى ال static design
يعنى كل system بيبقى فيه abstraction layer هيا layer بتعزل
ال layers الى فوق عن ال layers الى تحت ذی ال windows
بيعزل ال application عن ال hardware

عشان كذا ال design دا اسمه static design لانه
مبيتغيرش ... طول عمر ال DIO هتبقى موجوده فى
ال MCAL و ال LED فى ال HAL و هكذا



Testing note

لو عايز اعمل test على system ذی دا



1- هعمل test على module يعنى مثلا اجرب

اعمل call ل led_on و اشوف هل فعلا هتعمل

Call ل DIO_writeCh و تحط فى ال

Parameters ال ch_num الصح و رقم 1

ولا لا

ازای هعمل module test منغير ماعمل

?? #include DIO.h

انى اعمل حاجه اسمها **stup** معناها انی اعمل

Function اسمها DIO_writeCh() و اسيبها فاضيه ...

لو ال LED module عملها call فعلا بيقى ال test بتاع

ال led module كويس

و فى اخر ال test بنطلع test report فيه كل test و

اتعمل ازای و pass ولا fail

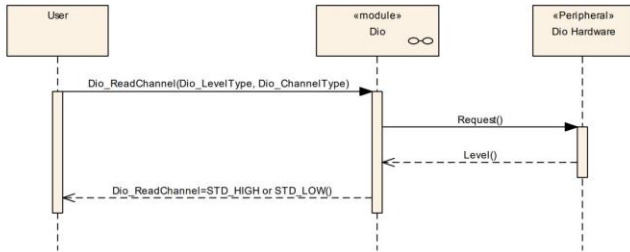
عيوب ال layer architecture

1- High over head

لان في function calls كثيره و دا بيضيع time كبير بسبب ال context switching

2- Large Code size

Sequence diagram



دا مثلا diagram بيوضح لو ال user عايز ي read channel بيعمل ايه ؟
بيبعث Dio_ReadChannel(Dio_LevelType, Dio_ChannelType)
لل Dio module و ال Dio module بيعت request و يرجع بال level بتاع ال pin

ال dashed line الى راجع دا معناه ان ال function بت return void

https://www.autosar.org/fileadmin/user_upload/standards/classic/19-11/AUTOSAR_SWS_DIODriver.pdf (page 43)

2- Dynamic design

الحاجات الى بتاثر على ال dynamic design

- 1- Tasks
- 2- Priorities
- 3- Timing(latency , response , jitter)
- 4- CPU load

هنا انا بشوف كل task هيتعملها call امتي
فكل task بتشتغل لما يحصل ال activation condition بتاعها ... ذي في ال RTOS
ذي مثلا

1- Event based task

بتستنى حد يعمل event معين

2- Interrupt based

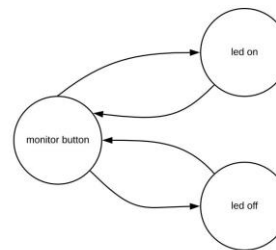
3- Periodic task / cyclic task / time based

بتشتغل مثلا كل ما ال time بتاعها يجي

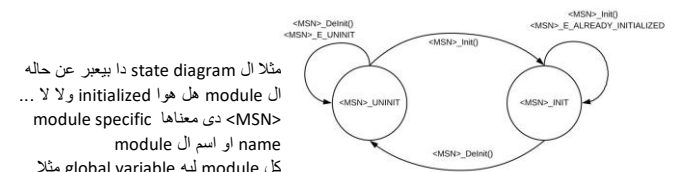
هنا مثلا انا الى بختار ال priority و ال tasks هيتعملها activation امتي
ال dynamic design ممكن نمثله بال

- 1- state diagram
- 2- sequence diagram

State diagram

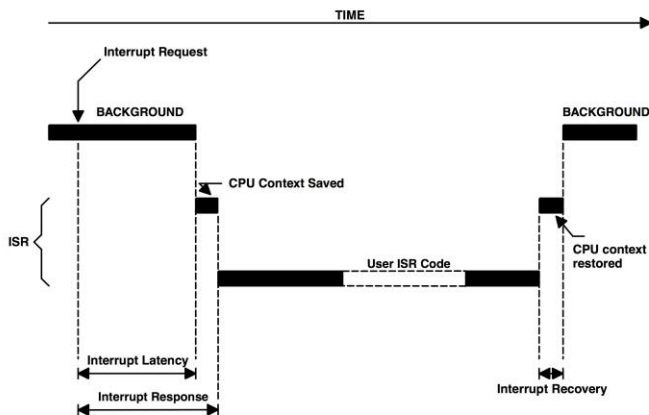


دا برضو diagram بيوضح العلاقة بين ال
functions و ال events و ال calls في ال
simple led / switch dynamic design
application



مثلا ال state diagram دا بيعبر عن حالة
ال module هل هو initialized ولا لا ...
<MSN> دي معناها module specific
name او اسم ال module
كل module ليه global variable مثلا
بالتما هينقي قيمته ب <MSN>_UNINIT لو
عملنا call لل <MSN>_DeInit() او هتبقى
قيمه <MSN>_INIT لو عملنا call لل
function الى اسمها <MSN>_Init()

interrupt latency , interrupt response



ال interrupt latency هوا الوقت الى بياخده ال microcontroller من ساعه ما يجي ال interrupt لغايه ما
بيبتدي ينفذ ال context switching
ال interrupt response الوقت الى بياخده من اول ما يحصل ال interrupt لغايه ما بيبتدي تنفيذ اول سطر في
ISR ال
ليه ممكن ال interrupt بتاخر ؟

- 1- ممكن ابقى شغال في ISR تاني اعلى priority (فدائما بنحاول نعمل optimization في ال ISR)
ليه ؟
a. عشان لو في interrupt ليه priority اقل ميستناش كثير
b. عشان اي interrupt يقطع اي task شغاله (دا في ال RTOS) فهاكل من وقت
ال task
2- لو انا عامل interrupts عشان critical section مثلا
a. فلان اقل كود ال critical section عشان اقل ال latency
و لو مضطر ازود ال critical section بضطر استخدم ال semaphores
كل ما ال latency يفتي اقل كل ما ال design يفتي احسن

CPU load

ال CPU قاعد فترة idle قد ايه يعني مفيش tasks شغاله ... كل ما ال CPU يفتي idle اكثر كل ما ال design
يفتي احسن ليه ؟
لان اكيد لو قاعد idle اي interrupt او task هتنتج هتشتغل على طول لان مفيش حاجة شغاله فبالتالي برضو ال
latency , response times
ال function calls الكثيره بتزود ال CPU load
لو ال task بيتعملها activation كثير (تاسك مثلا بيتعملها activation كل 1 ms) دا هيزود ال CPU load