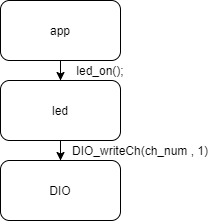
**AUTOSAR  
lecture 1**

**Testing note**

لو عايز اعمل test على system ذٍى 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

* ازاى هعمل module test منغير ماعمل

#include DIO.h ؟؟

انى اعمل حاجه اسمها stup معناها انى اعمل

Function اسمها DIO\_writeCh() و اسيبها فاضيه ... لو ال LED module عملها call فعلا يبقى ال test بتاع ال led module كويس

و فى اخر ال test بنطلع test report فيه كل test و اتعمل ازاى و pass ولا fail

1. **Static design**

ال static design هوا انى اقسم ال software بتاعى لاجزاء كل جزء مسؤل عن function معينه كدا ال system بقى مش complex و بقى اسهل فى ال testing , debugging بدل ما ابقى ب test , debug ال system كله على بعض

افتكر :

* ال module testing هوا انى بtest كل module لوحده ... بتاع ال timer لوحده و ADC لوحده و هكذا
* ال integration testing انى 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 يبص عالكود بتاعى عشان يتاكد من حاجات مينفعش يتعملها test ذى مثلا اسماء ال functions

ال static software design بيفيد فى ال reusability انى اقدر اجيب module كنت كاتبه قبل كدا و استخدمه تانى فى كل project

عشان كدا 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

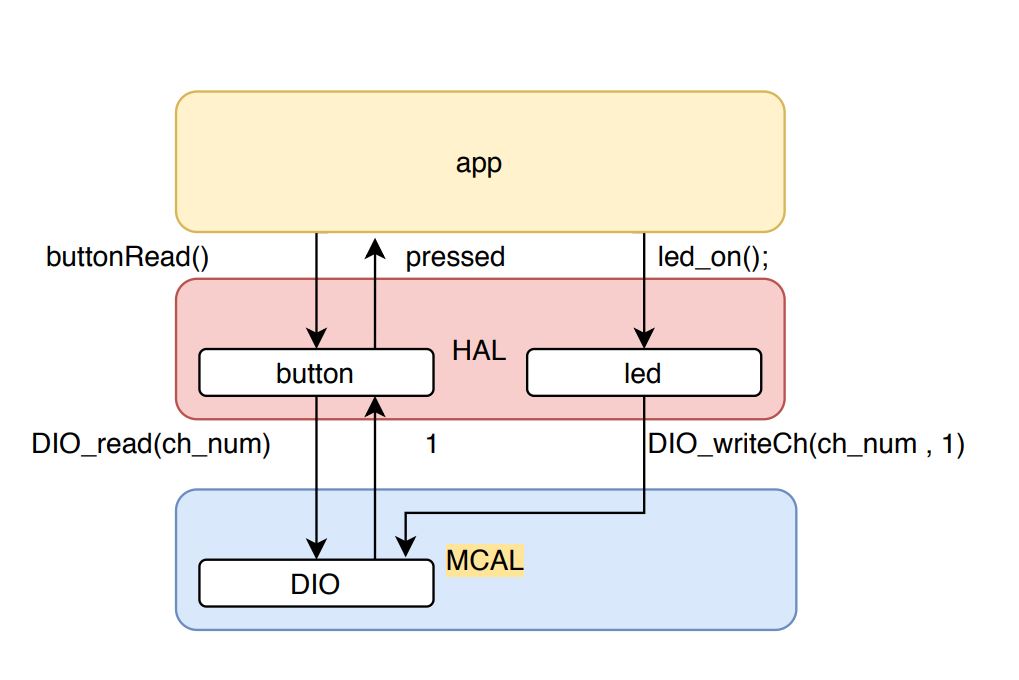
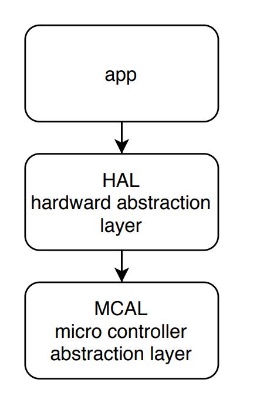
**Layers**

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

**Software design**

ال software design بيتقسم ل :

1. Static design
2. Dynamic design

****

**Software requirements**

هوا عباره عن شرح او اقتراح لل software architecture الى هتحقق ال functionality المطلوبه من ال software دا

طيب هيا ايه ال requirements دى ؟

1. Functional requirements

هيا requirements خاصه بال behavior او الtasks المطلوبه من ال SW دا

* 1. Positive requirements (useful function)

ذى ان لازم ال function دى لو دخلتلها ال input كذا هيطلع output كذا ... الى هوا الحاجه المطلوبه فعلا من ال function او ال software

* 1. Negative requirements

ان مثلا لو دخلتله input غلط هى report error او مينفذش حاجه معينه

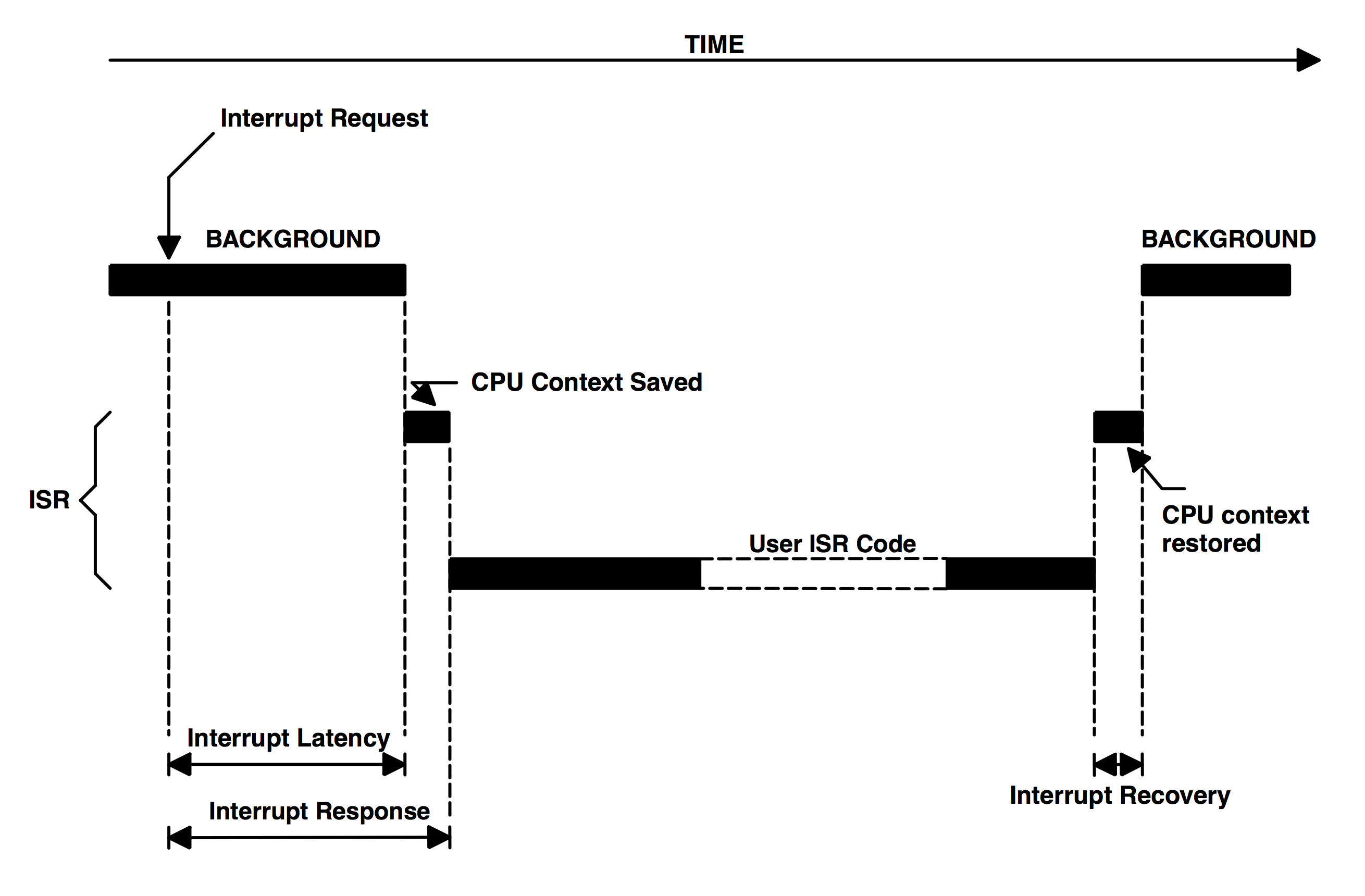
1. Nonfunctional requirements
   1. Real time constrains
   2. Resource constrains

ذى انى مضطر اشتغل على controller معين

1. Other requirements

ذى انى لازم مثلا اكتب بال autosar standards

**interrupt latency , interrupt response**

****

ال interrupt latency هوا الوقت الى بياخده ال microcontroller من ساعه ما يجى ال interrupt لغايه ما يبتدى ينفذ ال context switching

ال interrupt response الوقت الى بياخده من اول ما يحصل ال interrupt لغايه ما يبتدى تنفيذ اول سطر فى ال ISR

ليه ممكن ال interrupt يتاخر ؟

1. ممكن ابقى شغال فى ISR تانى اعلى priority (فدايما بنحاول نعمل optimization فى ال ISR )

ليه ؟

* 1. عشان لو فى interrupt ليه priority اقل ميستناش كتير
  2. عشان اى interrupt بيقطع اى task شغاله (دا فى ال RTOS ) فهاكل من وقت ال task

1. لو انا عامل disable لل interrupts عشان critical section مثلا
   1. فلازم اقلل كود ال critical section عشان اقلل ال latency

و لو مضطر ازود ال critical section بضطر استخدم ال semaphores

كل ما ال latency يبقى اقل كل ما ال design يبقى احسن

**CPU load**

ال CPU قاعد فتره idle قد ايه يعنى مفيش tasks شغاله ... كل ما ال CPU يبقى idle اكتر كل ما ال design يبقى احسن ليه ؟

لان اكيد لو قاعد idle اى interrupt او task هتيجة هتشتغل على طول لان مفيش حاجه شغاله فبالتالى برضو ال latency , response times اقل

ال function calls الكتيره بتزود ال CPU load

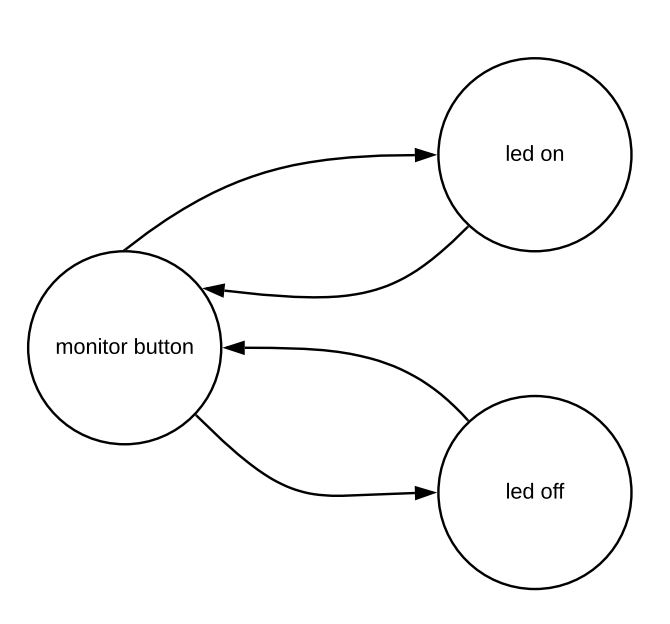
لو ال task بيتعملها activation كتير ( تاسك مثلا بيتعملها activation كل 1 ms ) دا هيزود ال CPU load

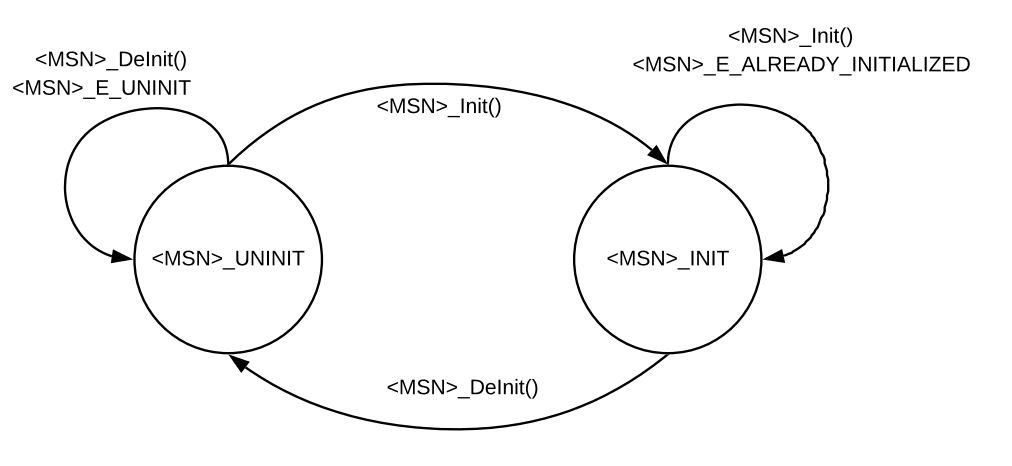
مثلا ال state diagram دا بيعبر عن حاله ال module هل هوا initialized ولا لا ...

<MSN> دى معناها module specific name او اسم ال module

كل module ليه global variable مثلا يائما هيبقى قيمته ب <MSN>\_UNINIT لو عملنا call لل<MSN>\_DeInit() او هتبقى قيمته <MSN>\_INIT لو عملنا call لل function الى اسمها <MSN>\_Init()

**State diagram**





1. **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 امتى  
فكل task بتشتغل لما يحصل ال activation condition بتاعها ... ذى فى ال RTOS كدا

ذى مثلا

1. Event based task

بتستنى حد يعمل event معين

1. Interrupt based
2. Periodic task / cyclic task / time based

بتشتغل مثلا كل ما ال time بتاعها يجى

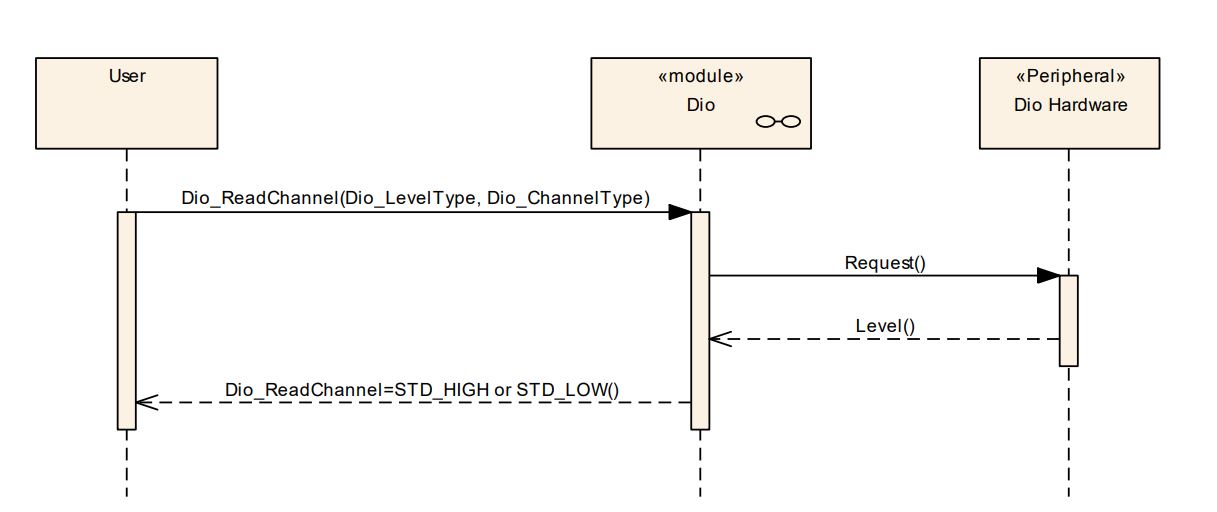
هنا مثلا انا الى بختار ال priority و ال tasks هيتعملها activation امتى

ال dynamic design ممكن نمثله بال

1. state diagram
2. sequence diagram

دا برضو diagram بيوضح العلاقه بين ال functions و ال events و ال calls فى ال dynamic design ل simple led / switch application

**Sequence diagram**



دا مثلا diagram بيوضح لو ال user عايز ي read channel بيعمل ايه ؟

بيبعت Dio\_ReadChannel(Dio\_LevelType, Dio\_ChannelType)لل DIO module و ال dio module يبعت request و يرجع بال level بتاع ال pin

ال dashed line الى راجع دا معناه ان ال function بتreturn void

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

**عيوب ال layer architecture**

1. High over head

لان فى function calls كتيره و دا بيضيع time كبير بسبب ال context switching

1. Large Code size