# DESIGN4: TIMING DIAGRAMS

### TODAY

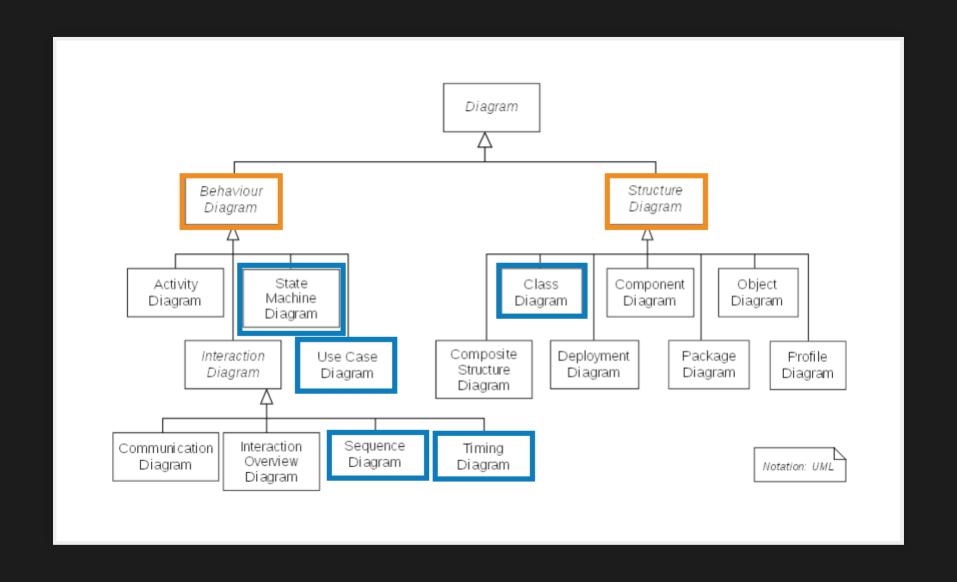
- Timing diagrams
  - Sequence Diagrams
  - TDD
- PWM and Peripherals in micros recap

#### UML MODELING1 RECAP

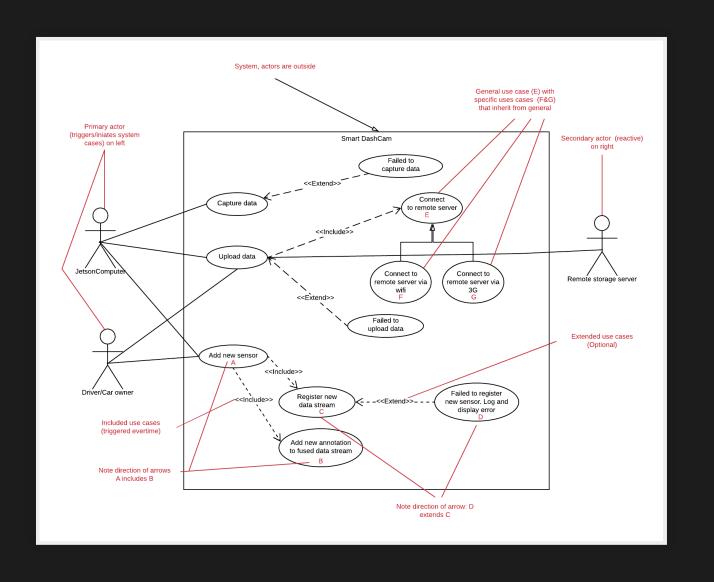
- UML is a standard specifying the syntax and semantics of graphical design languages.
  It covers amny different design diagram typs
- Class diagrams as primarily used for Object Orientated software design
  - Design patterns are an extremely useful tool for designing OO software
- Use case diagrams: completed example from smart dash cam class exercise available in gslides:
  - For more examples: https://www.uml-diagrams.org/use-case-diagrams-examples.html
  - Smart Dash Cam https://tinyurl.com/y66sala8

### MODELING1 RECAP

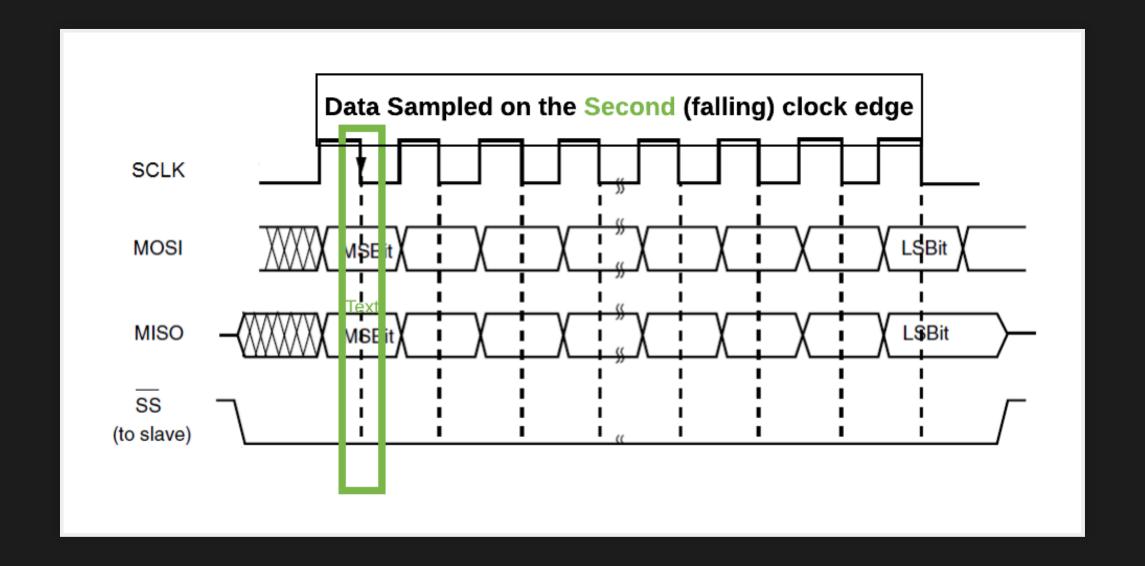
#### DIAGRAMS: MANY SYNTAXES



## MODELING1 RECAP: USE CASE DIAGRAMS

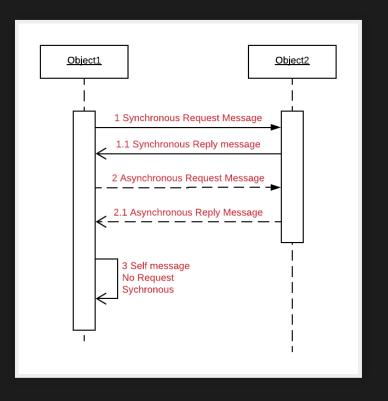


### TIMING DIAGRAMS

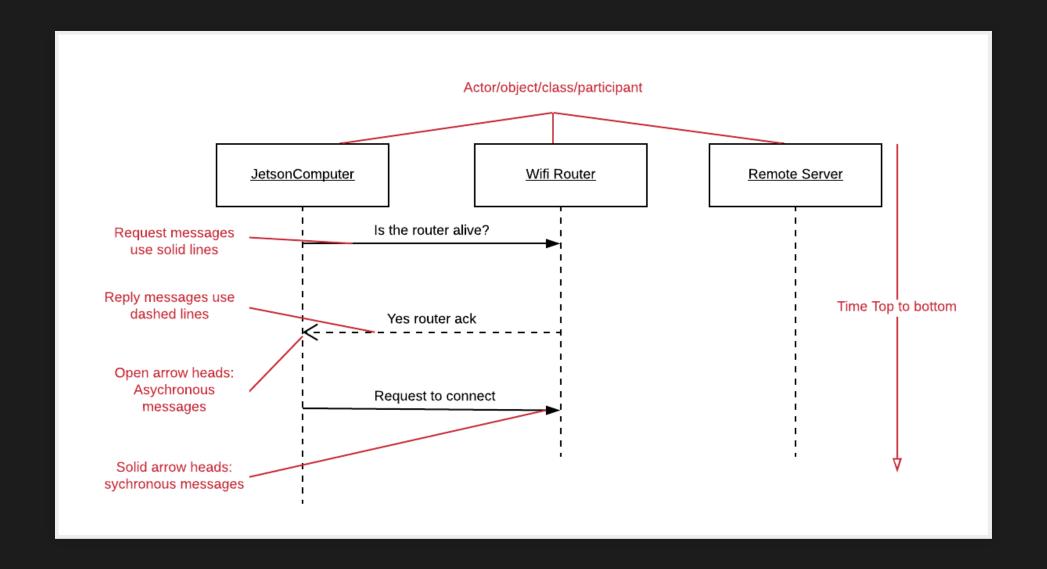


## SEQUENCE DIAGRAMS

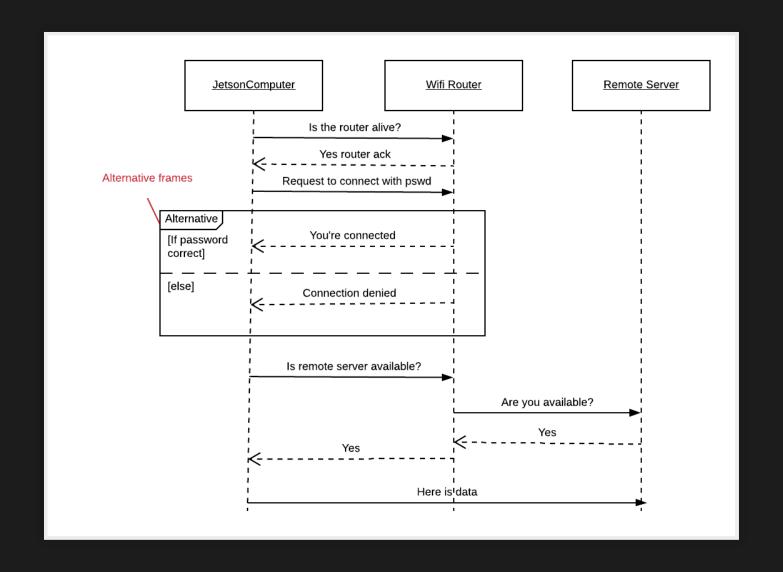
- UML used to show how objects/classes/participant interact via messages through time:
  The order/sequence of events
- Used to model systems abstractly and in code design



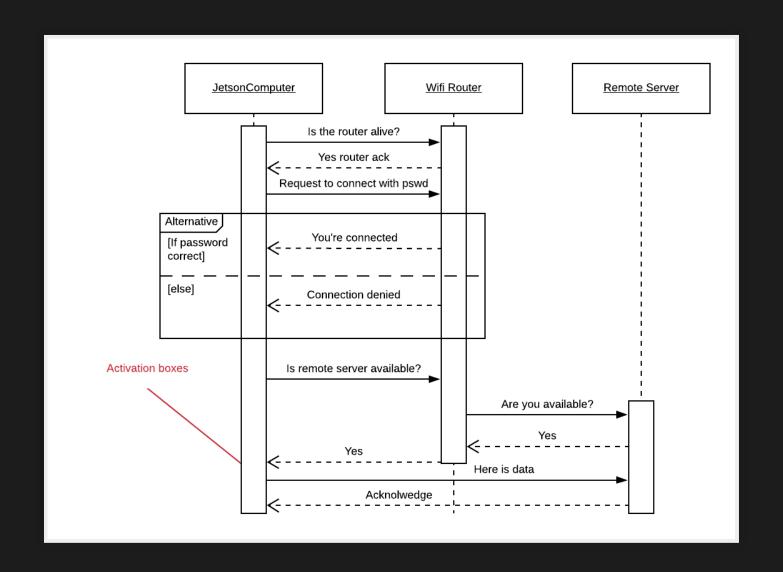
## SEQUENCE DIAGRAMS



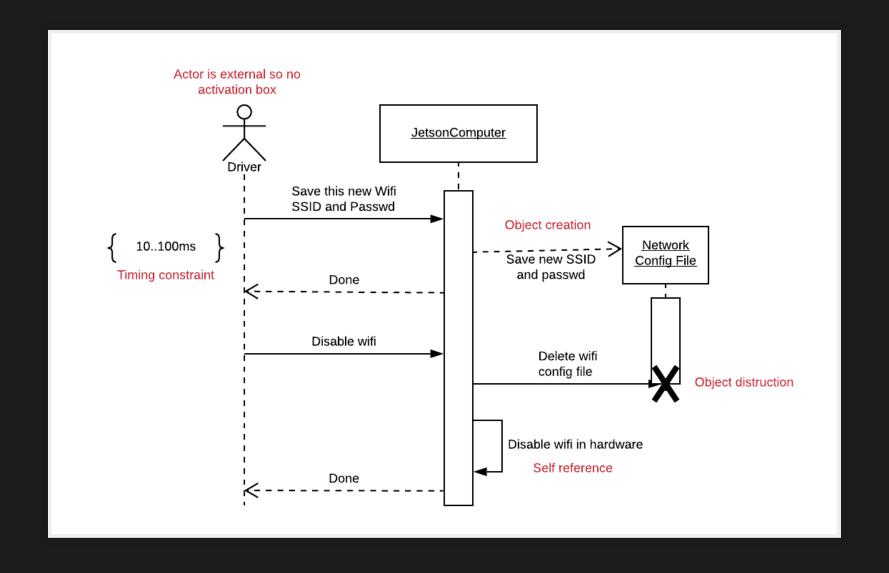
## SEQUENCE DIAGRAM



## SEQUENCE DIAGRAM



## SEQUENCE DIAGRAM

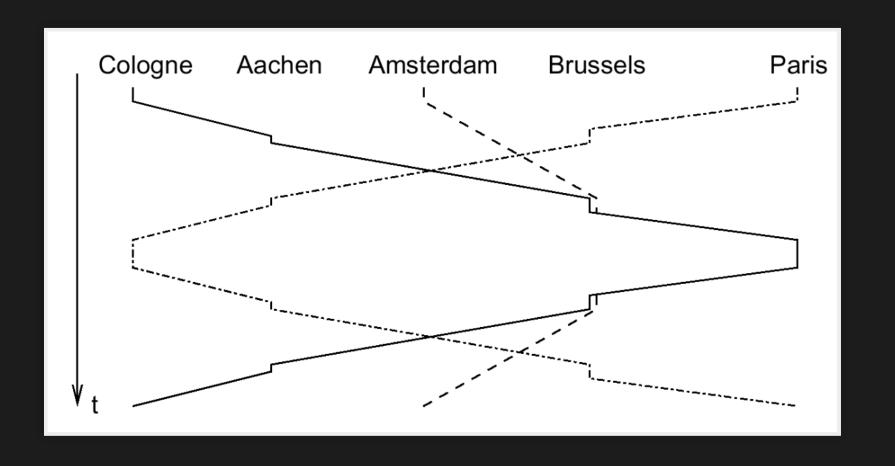


### SEQUENCE DIAGRAM SYNTAX SUMMARY

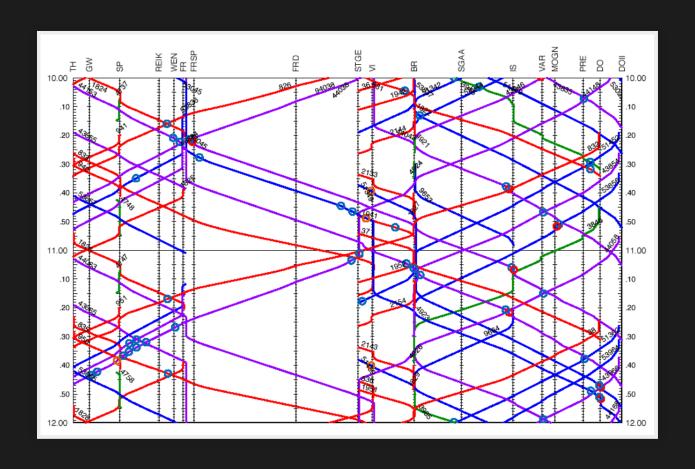
- Actors and Objects(Could be classes) are drawn from left to rights
- lifelines/timelines show time vertically downwards
- Messages (could be functions in classes): sent from caller to receiver
  - Solid (Request message)
  - Dashed (Reply message)
  - Synchronous (blocking, response expected, solid arrow)
  - Asynchronous (none blocking, response not expected in a given time) open arrow
  - Can number them for ease
- Alternative frames (choice) boxes. If only (no else) boxes and loops look the same without the dashed line
- Activation boxes: When and how long an object is performing/idle
- Add precise timing constraints in curly brackets

## TIME/DISTANCE DIAGRAMS (TDD)

- These diagrams are more concise than sequence diagrams, showing only time that actions take in relation to others.
- Slope of lines shows rate (steep is slow, shallow is fast)

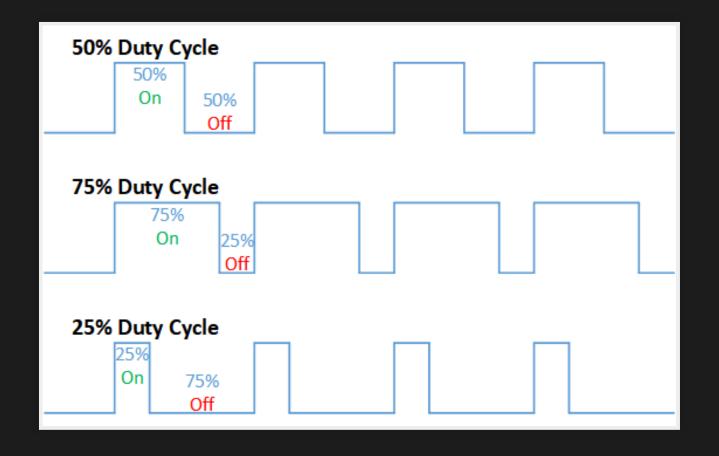


# TIME/DISTANCE DIAGRAMS (TDD)



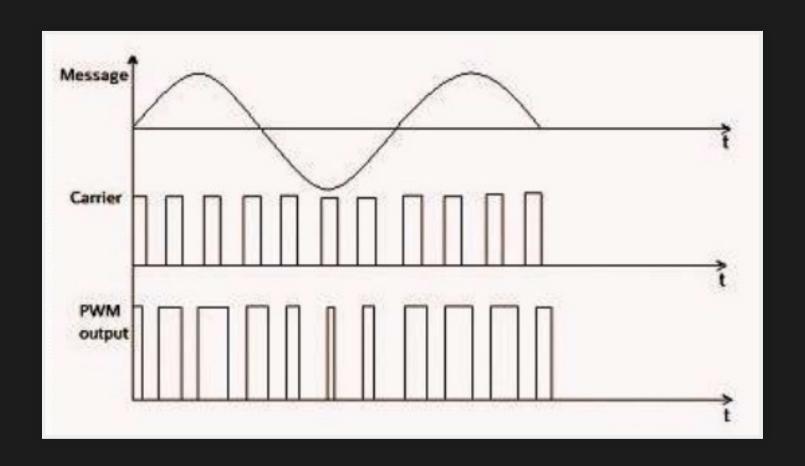
#### PULSE WIDTH MODULATION RECAP

 Most simple instance: a means of changing the duty cycle of a pulse signal so as to change the net power, eg dim a LED.



### PULSE WIDTH MODULATION RECAP

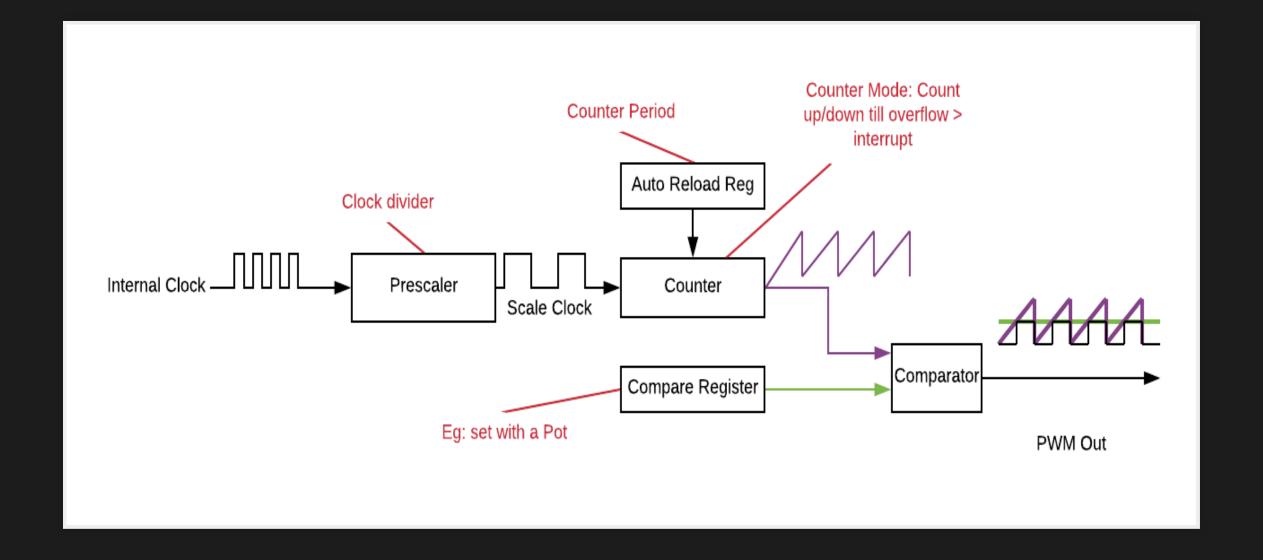
■ Can also use PWM to transmit information over a carrier wave



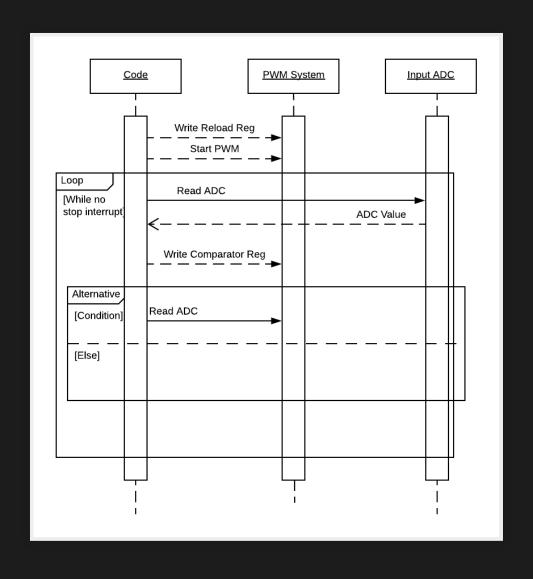
#### PULSE WIDTH MODULATION RECAP: MICROCONTROLLERS

- Microcontroller PWM options:
  - Use dedicated peripheral hardware
    - Flexible, minimal jitter, limited pins
  - Use software on a GPIO pin
    - Flexible, jitter subject to software limits (eg kernel schedler, execution speed), available on an GPIO pin
  - Pi also as a means of generating PWMs via DMA (more on that later this week)

### PULSE WIDTH MODULATION RECAP: MICROCONTROLLERS



## PWM SOFTWARE SEQUENCE DESIGN DIAGRAM



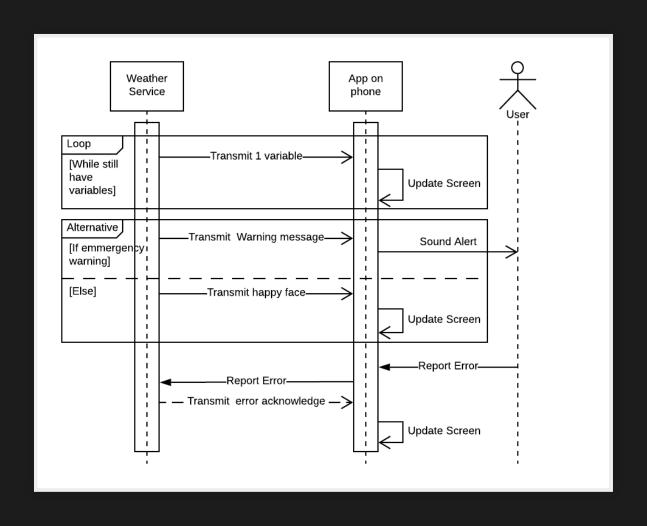
### SEQUENCE DIAGRAMS EXERCISE:

- Task: Draw a sequence diagram showing the weather service interacting with a user and a phone app. Show the following situations:
- The weather service sending a sequence of messages 1 at a time each containing 1 variable from the list: [Temperature, Humidity, Wind speed, Rainfall, Cloud Cover] to a weather app on a users phone.
- The weather service should check if it needs to send an emergency warning each time it completes sending the full list and if so the phone app should sound an alert. If not, send a smiley.
- The phone app should always display the latest message received
- After one instance of the above steps, the end the user should submit an error to the weather service via the phone app and receive an acknowledgement message in the app.

### SEQUENCE DIAGRAMS EXERCISE SYNTAX REMINDER:

- Actors and Objects(Could be classes) are drawn from left to rights
- lifelines/timelines show time vertically downwards
- Messages (could be functions in classes): sent from caller to reciever
  - Solid (Request message)
  - Dashed (Reply message)
  - Can number them for ease
  - Sychronouse (wait for response, solid arrow)
  - Asynchronouse (no response expected) open arrow
- Alternative frames (choice) boxes. If only (no else) boxes and loops look the same without the dashed line
- Activation boxes: When and how long an object is performing/idel
- Add precise timing constrainst in curly brackets

## SEQUENCE DIAGRAMS EXERCISE:



### HOMEWORK (NEXT LECTURE PREWORK)



## REFERENCES