



Project Structure

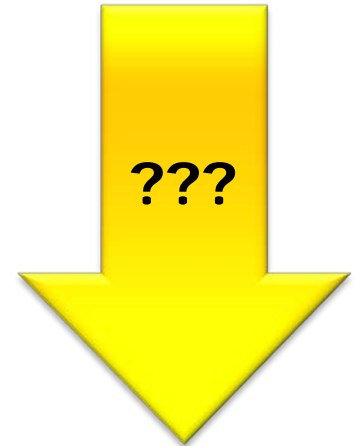
"Structure is everything"

Prof. Erich Styger
erich.styger@hslu.ch
+41 41 349 33 01

**Scriptum:
Project Structure**

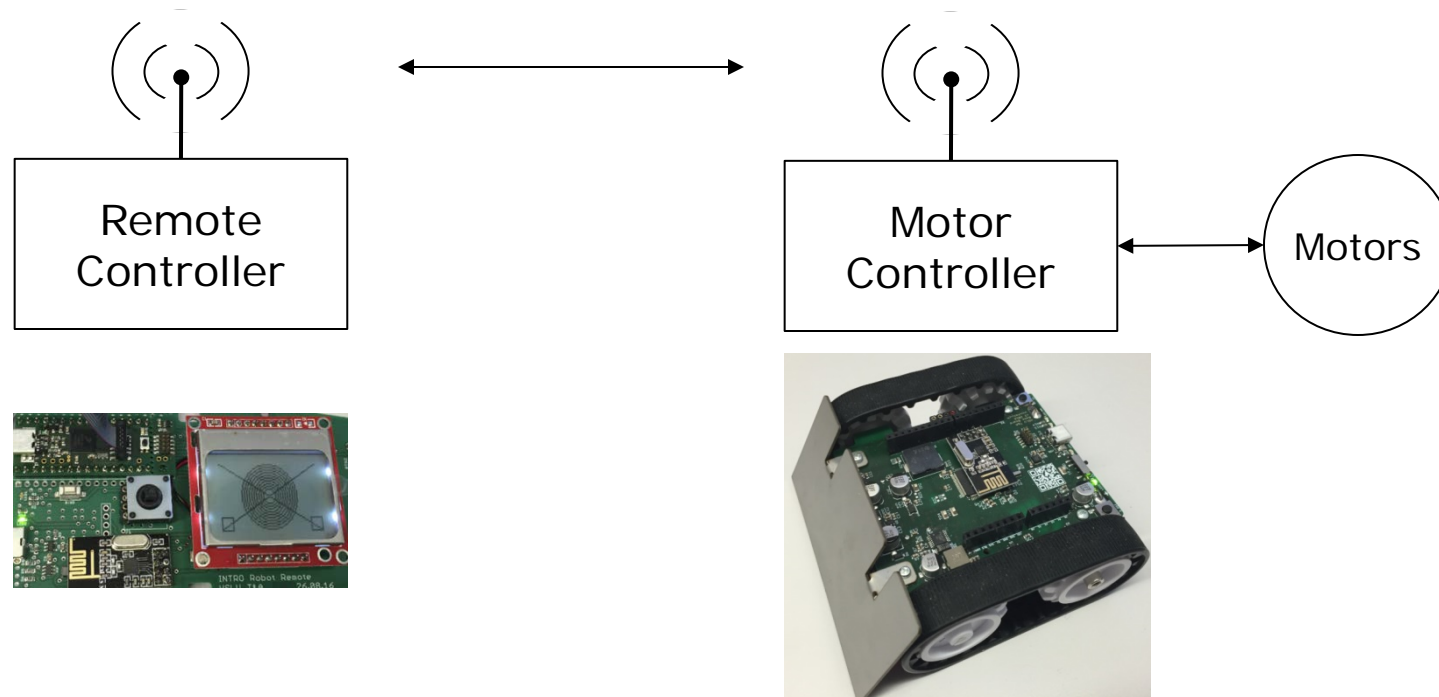
Learning Goals

- Problem: Requirements and Project Organization
- Requirements decomposition
- Top-Down approach
- Hardware blocks
- Project/Directory Structure

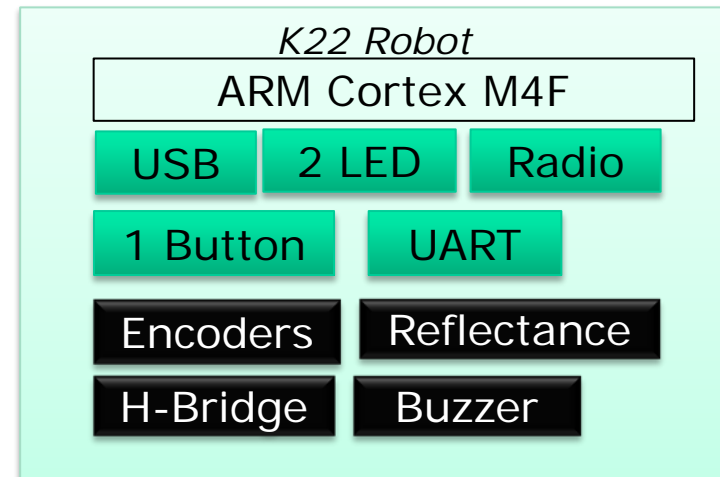
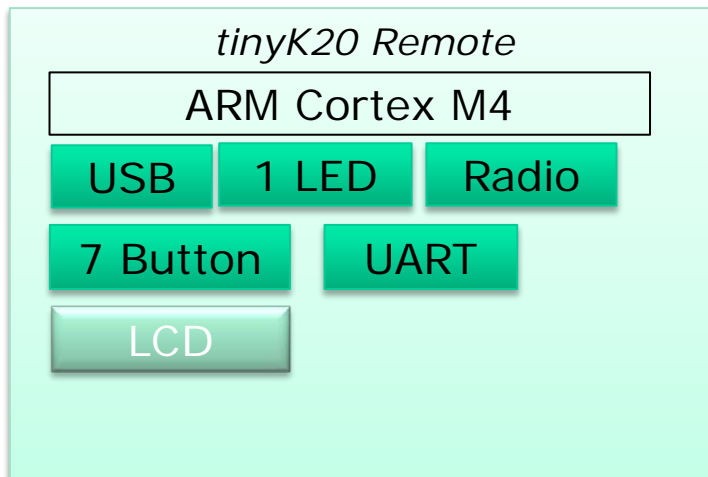


High Level Requirement

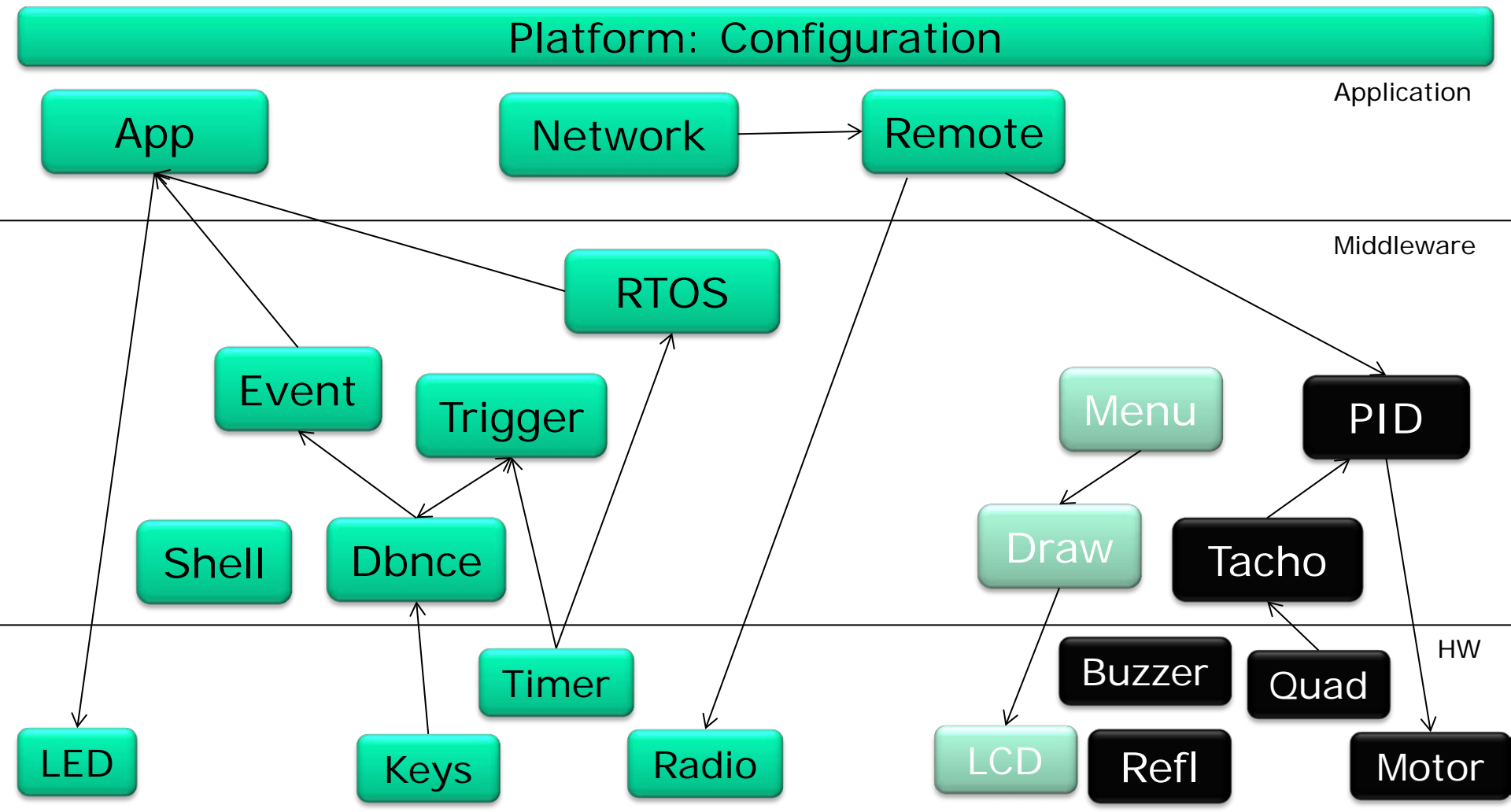
- Using FRDM-KL25Z/tinyK20 as remote controller
- Robot is executing commands and exploring environment



System Blocks: many in common!

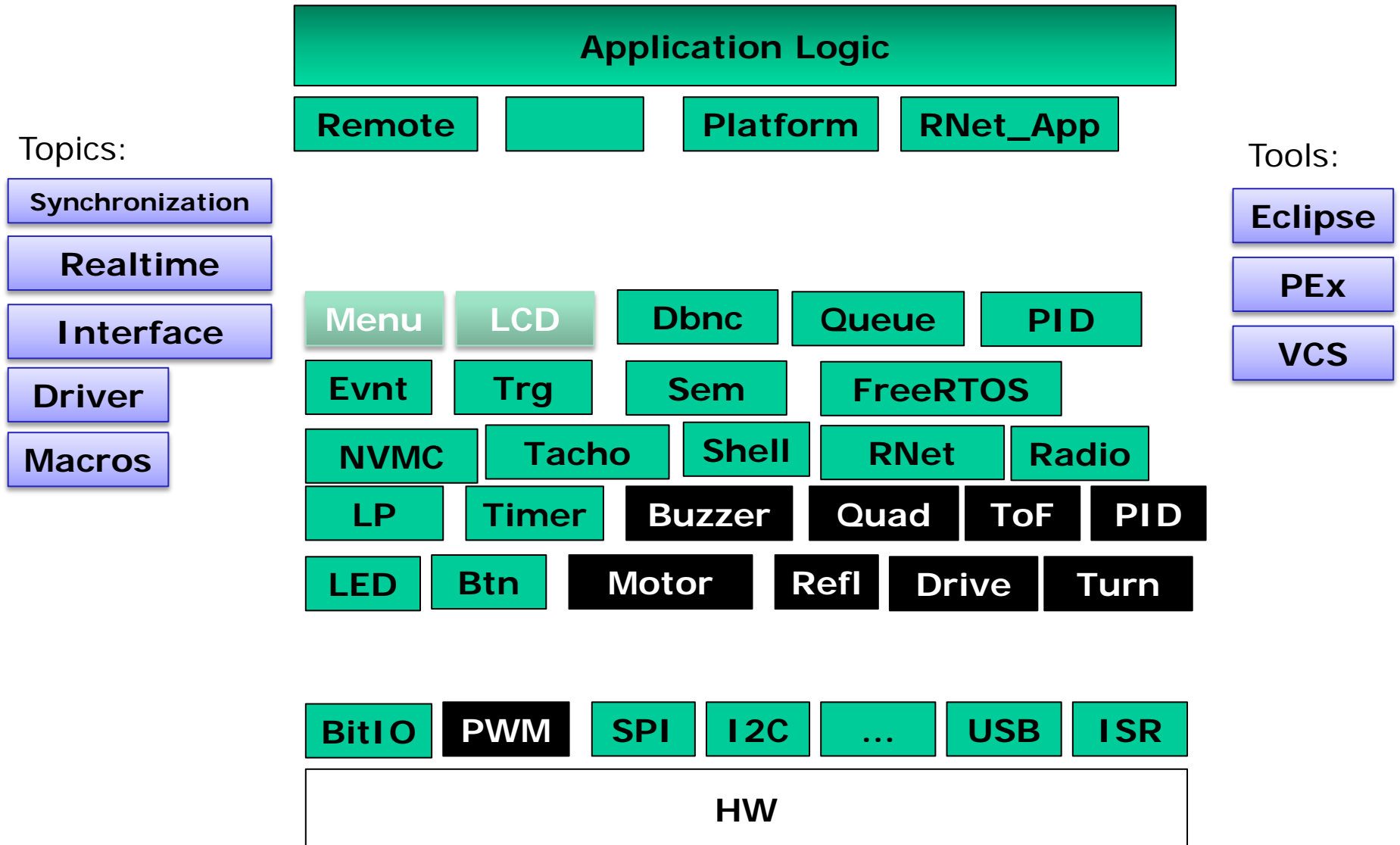


High Level System Architecture



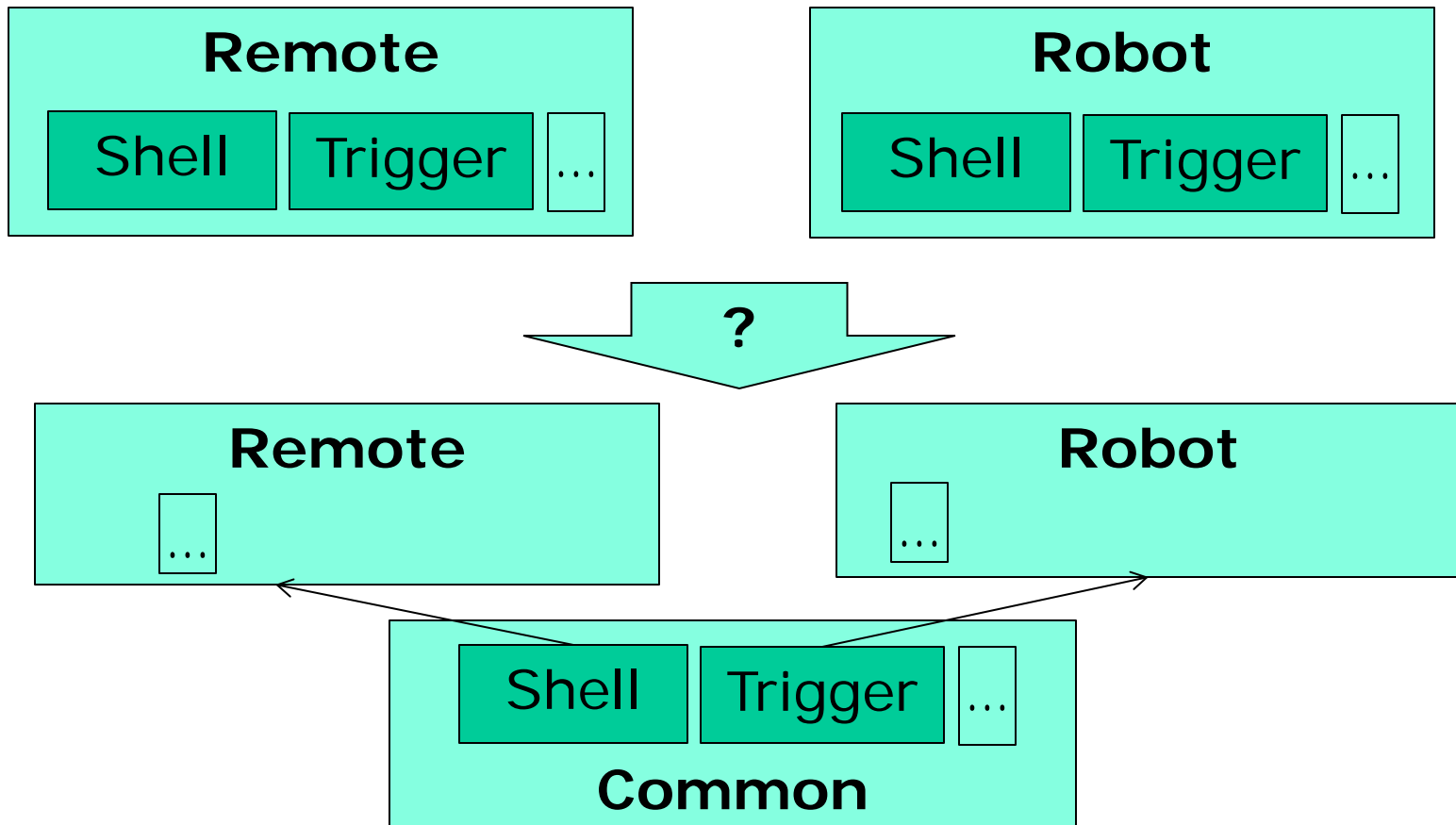
Processor Expert: Hardware & Low Level

Hardware/Software Layers



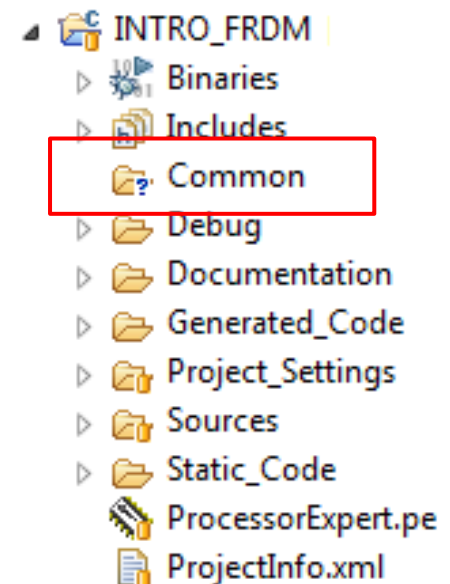
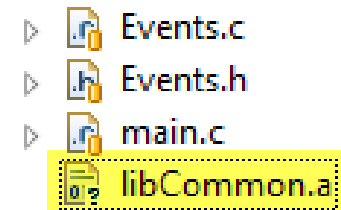
Sharing Common Files

- 'Library' of common files
- Different 'platforms': high end, low end, features, ...

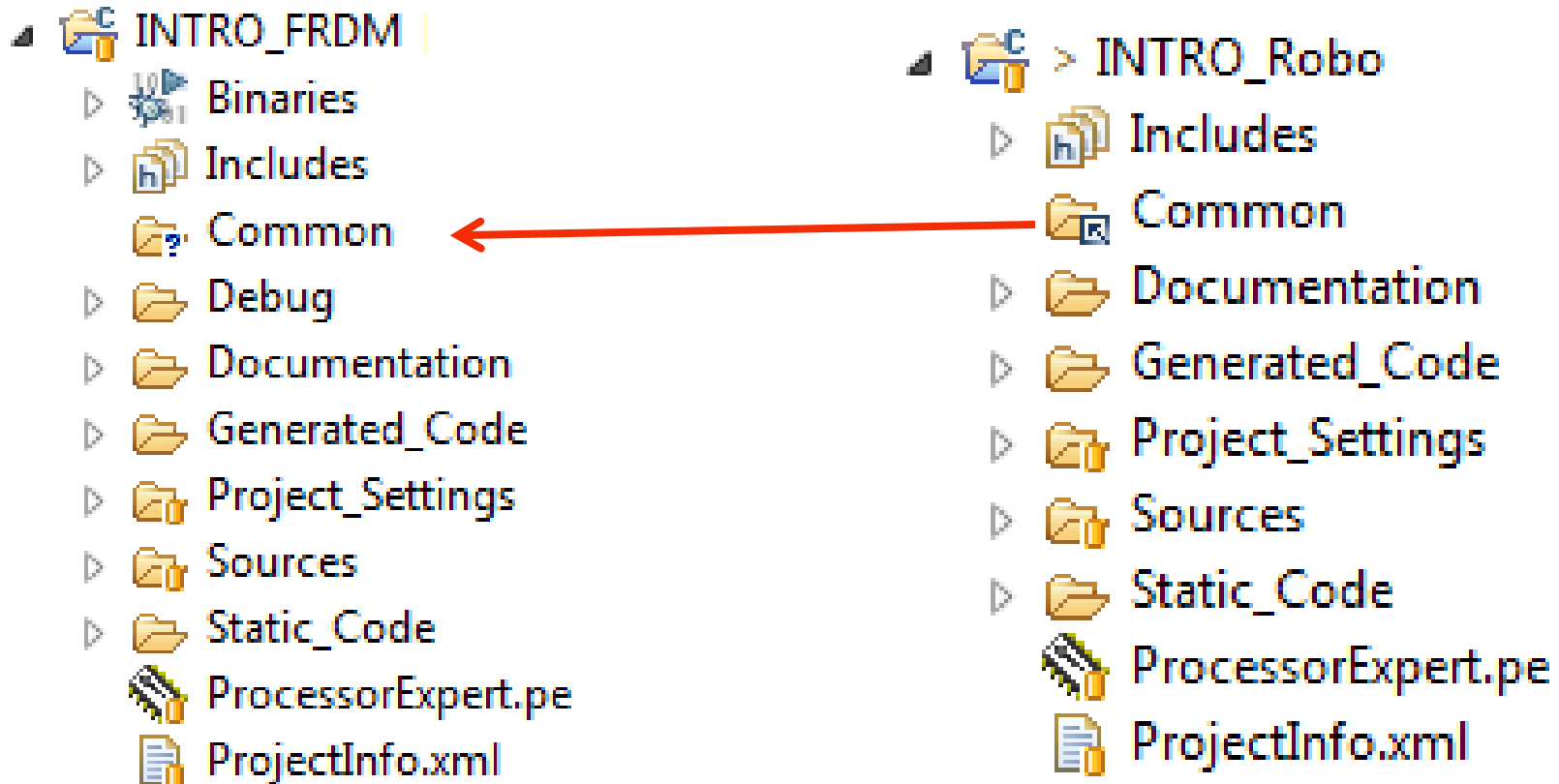


Eclipse: Shared Resources

- 'Real' Library (.a, .lib)
 - Good for 'true' library
 - 'static', cannot configure at run time
- Folder outside project
 - Where?
 - Eclipse limitation (project view)
- Folder inside one project
 - In which project?
 - Using Project References?
 - What to use?
 - (Virtual Group with links)
 - Linked Folder

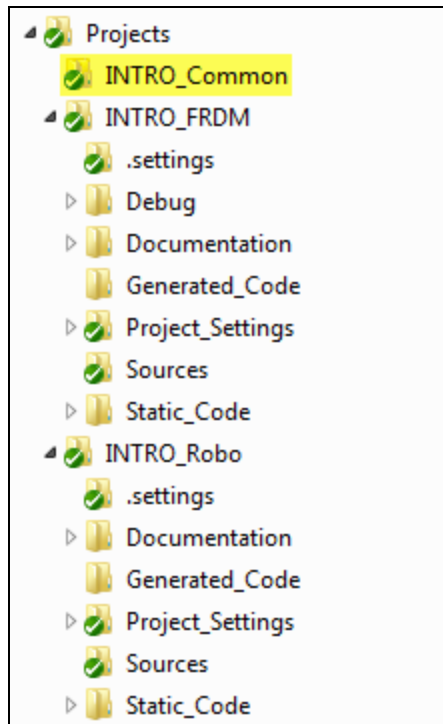


Linked Folder

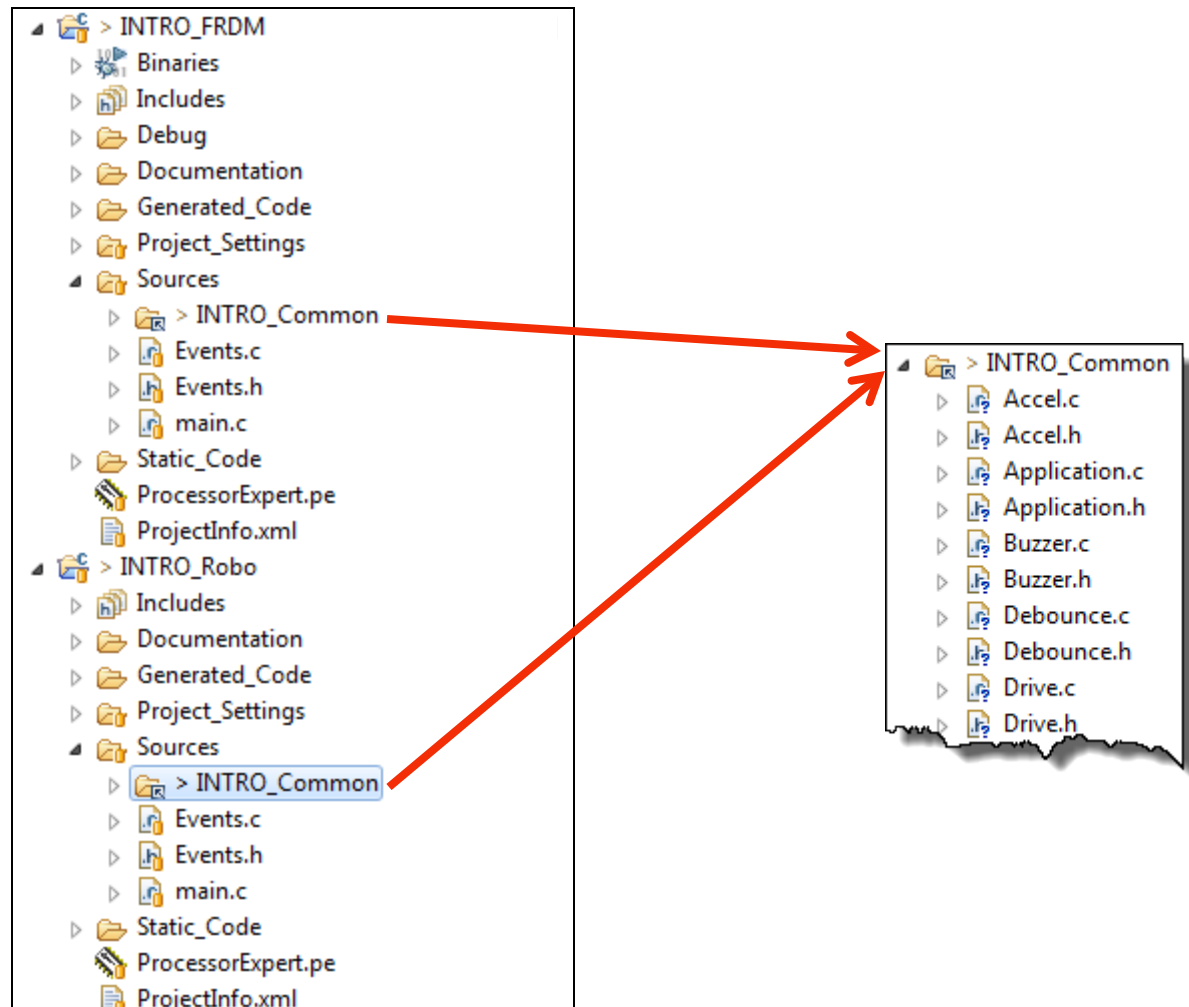


Common Folder with Links

Disk:

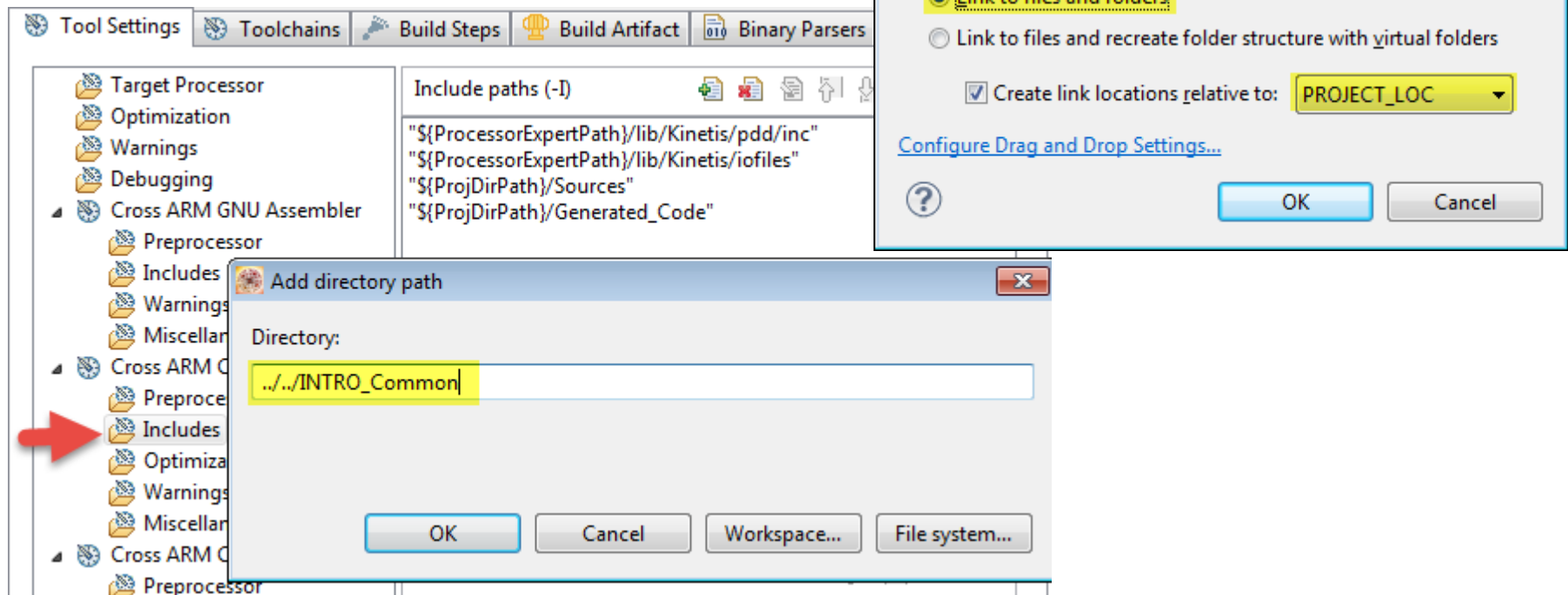


Eclipse:



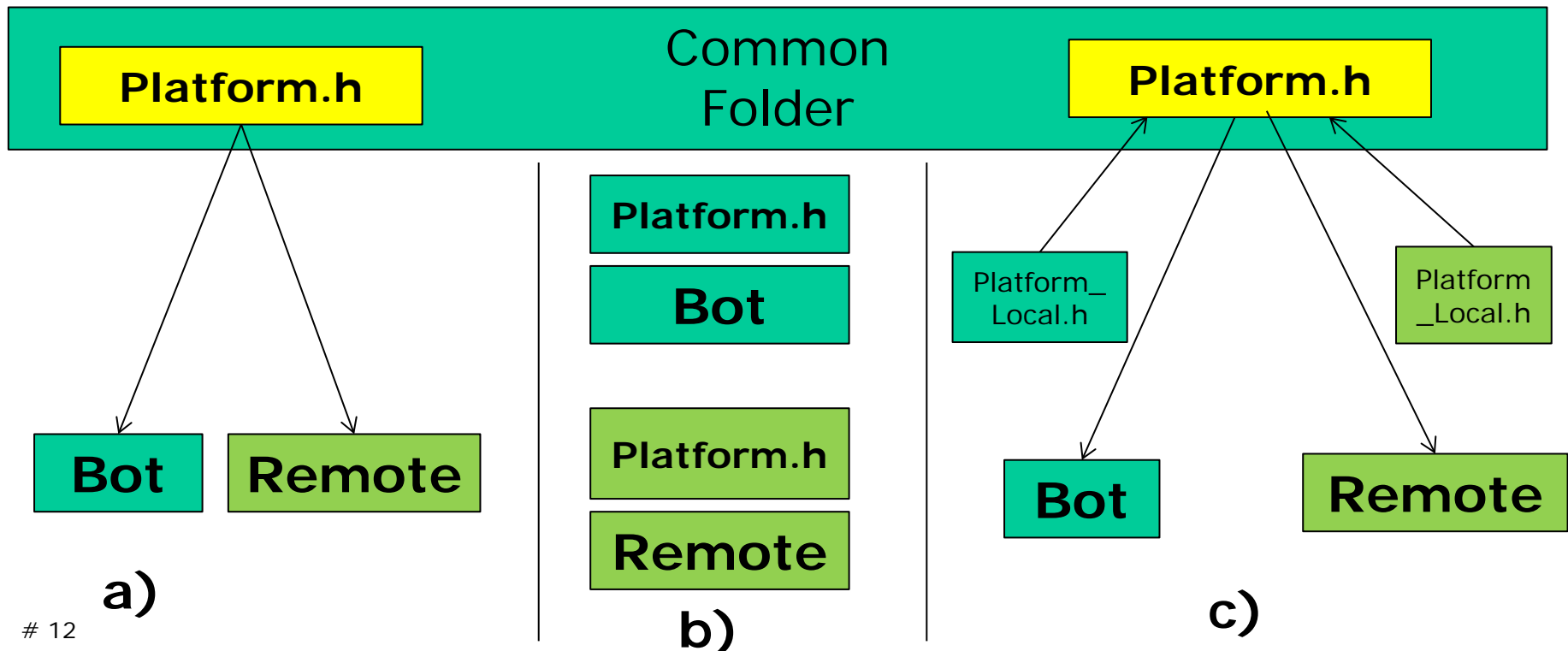
Eclipse: Linked Folder

- Instead of files, the folder is linked
- Drag & drop with CTRL key pressed
- **Add path for build tools!**
 - Relative to output folder!



Platform.h

- A way to configure 'Platform' or 'Product'
- Platform.h
 - Shared among projects: need to know 'who am I?'
 - Duplicated in each project: need to sync
 - Mixed approach with local/global platform configuration



INTRO Common Library Structure

- Common drivers in **INTRO_Common** (linked folder)
- Drivers guarded by PL_CONFIG macro
 - **#if PL_CONFIG_HAS_LEDS**
- Platform.h maps dependencies
 - **#define PL_CONFIG_HAS_BUZZER (1 && !defined(PL_LOCAL_CONFIG_HAS_BUZZER_DISABLED) && **PL_CONFIG_BOARD_IS_ROBO****
- Platform_Local.h can turn off functionality and defines board
 - **#define PL_LOCAL_CONFIG_HAS_LEDS_DISABLED**
 - **#define PL_LOCAL_CONFIG_BOARD_IS_FRDM (1)**



Board Identification

- Project Level (project settings)
- Compiler Define option (Preprocessor Macros)
 - `-dPL_BOARD_IS_REMOTE`
 - `#define PL_BOARD_IS_REMOTE`
 - `-dPL_BOARD_IS_ROBO=1`
 - `#define PL_BOARD_IS_ROBO 1`

```
#ifdef PL_BOARD_IS_REMOTE
    #define PL_NOF_LEDS 1
#elif defined(PL_BOARD_IS_ROBO)
    #define PL_NOF_LEDS 2
#else
    #error „Unknown board?“
#endif
```

Compiler Preprocessor Options

- D compiler option (same as #define in source)



The screenshot shows the 'Tool Settings' window of an IDE. The 'Toolchains' tab is selected. In the left-hand tree view, the 'Cross ARM C Compiler' is expanded, and the 'Preprocessor' sub-item is highlighted with a yellow background. On the right-hand side, there are two checkboxes: 'Do not search system directories (-nostdinc)' and 'Preprocess only (-E)', both of which are currently unchecked. Below these checkboxes is a section titled 'Defined symbols (-D)' with a toolbar containing icons for adding, removing, and editing symbols. A single symbol, 'PL_BOARD_IS_FRDM', is listed in the table below the title bar.



Defined symbols (-D)	
PL_BOARD_IS_FRDM	

Eclipse Indexer

- <http://mcuoneclipse.com/2012/03/20/fixing-the-eclipse-index/>

```
#if PL_IS_FRDM
    #define PL_NOF_LEDS      3
    /*!< FRDM board has 3 LED
#elif PL_IS_ROBO
    #define PL_NOF_LEDS      2
    /*!< We have 5 LED's on the SRB board */
#else
    #error "unknown configuration?"
#endif
```

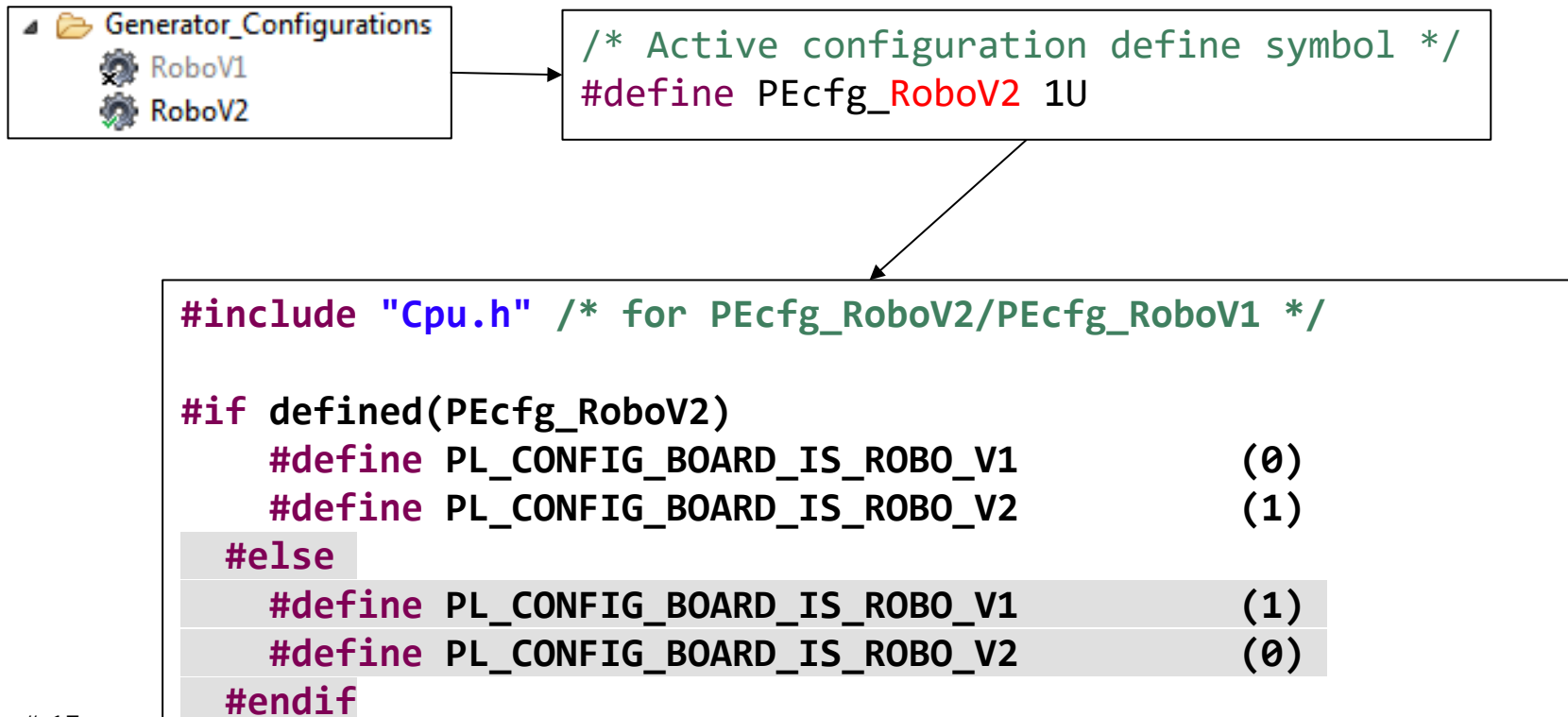
	Undo Typing	Ctrl+Z
	Revert File	
	Save	Ctrl+S
	Open Declaration	F3
	Open Type Hierarchy	F4

 Platform.h 

```
20 #define PL_IS_ROBO (defined(PL_BOARD_IS_ROBO) || defined(PL_BOARD_IS_INTRO_FRDM) || defined(PL_BOARD_IS_INTRO_ROBO))
21
22
```


Processor Expert Configuration

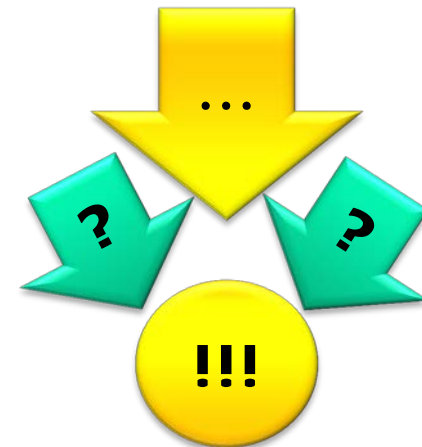
- Multiple named configurations
- One active at a time
- In configuration components can be enabled/disabled
- Configuration creates define in CPU.h



Summary

- Problem: Requirements and Project Organization

- Requirements (high level)
- System Decomposition
- System Overview
- Directory Structure
- Project Organization
- Possibility of preprocessor define
- Linked Folders
 - Compiler Include Paths



Lab: Project Structure

- Get INTRO_Common Library
- Copy Platform_Local.h
- Copy Processor Expert components
 - or: copy .pe file
- Inspect
 - Drivers
 - Platform.h
 - Platform_Local.h

