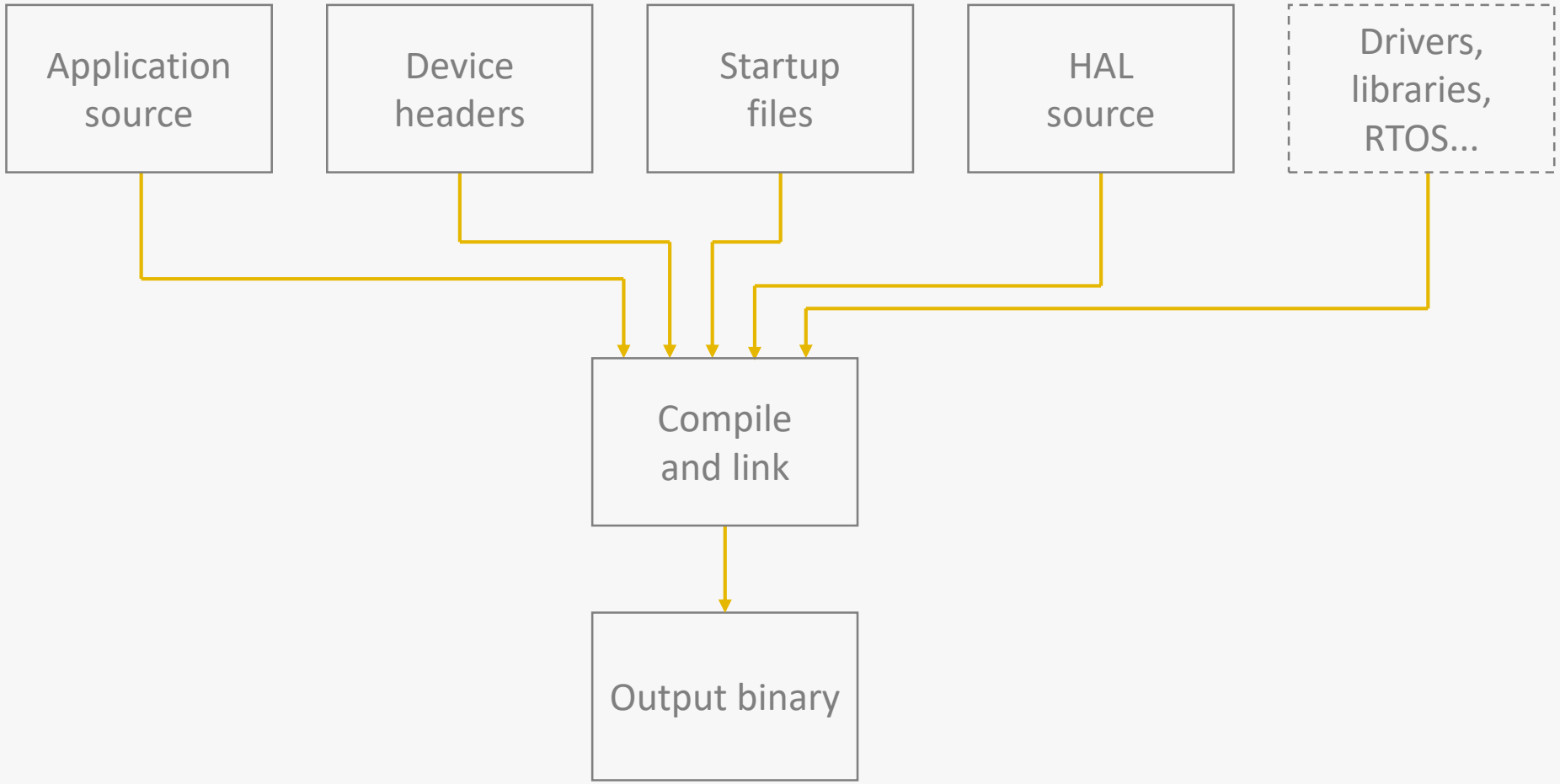
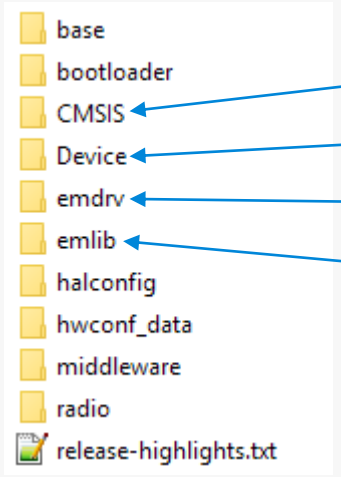
EFM32 Series 0: Tools and IDEs

Building Software



SDK Contents

\platform



Generic Cortex header files

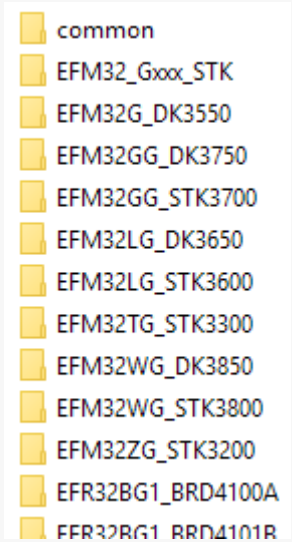
Device specific header files

Driver libraries provided by Silicon Labs

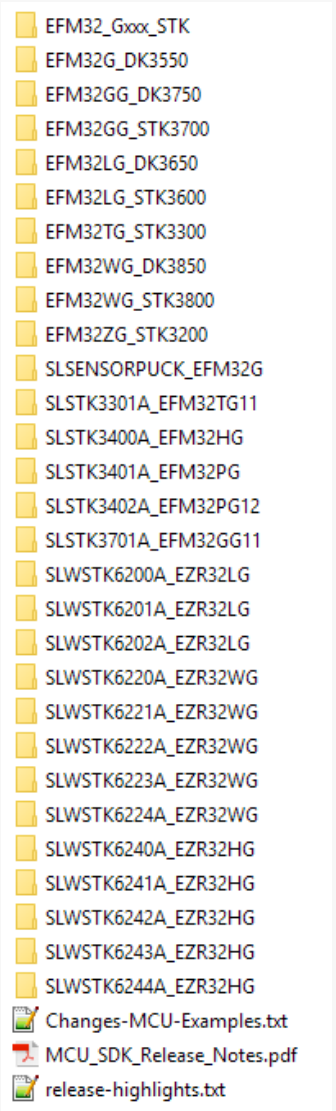
EFM32 HAL

Kit-specific files: BSP, firmware, examples →

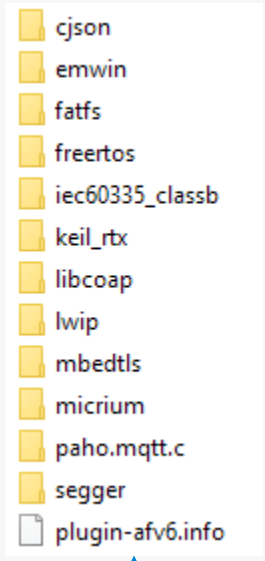
\hardware\kit



\app\mcu_example

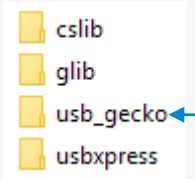


\util\third_party



Third party libraries. RTOS, emWin ...

\platform\middleware



EFM32 USB Stack

Chip Configuration – Startup Files

```
startup_efm32gg.c
startup_efm32gg.s
```

- Defines the interrupt vector and reset handler.
- Contains default while(1) implementations of ISRs that are weakly linked
- Toolchain specific
  - Assembly version for all toolchains
  - C version available for IAR
- EFM32 family specific

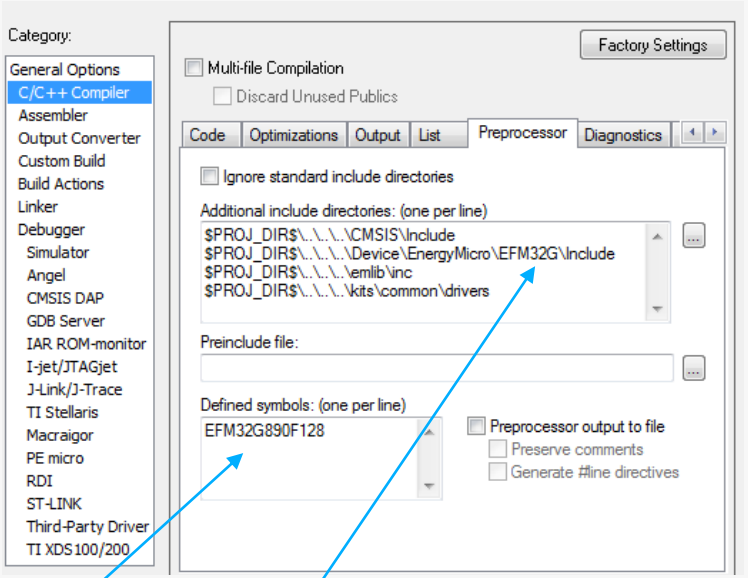
```
system_efm32gg.c
```

- Contains system functions
- EFM32 family specific

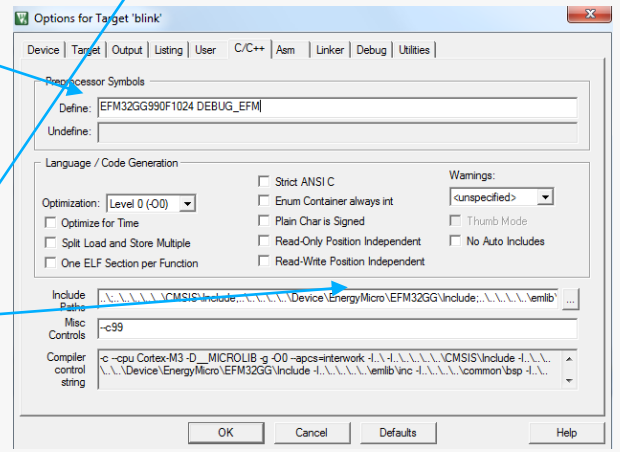
Chip Configuration – Device Headers

- Device headers declare addresses and bit-fields of all peripheral registers
- Include path is family specific
  - E.g. Device/SiliconLabs/EFM32GG/Include
- All files include em_device.h
- Device specific definition must be set in preprocessor
  - For GCC: -DEFM32GG990F1024

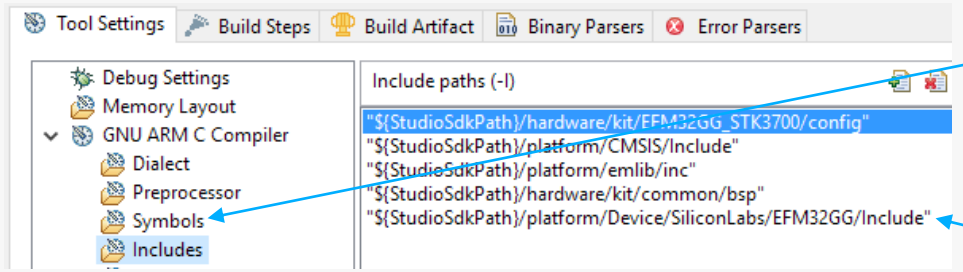
IAR



Keil



Simplicity IDE



Device specific preprocessor definition

Family specific include directory

Chip Configuration – Linker

- Linker files define memory sizes (Flash/RAM) and placement
- Toolchain specific
- For GCC
  - Linker files provided under Device/EnergyMicro
  - Note: only max size for family defined
- IAR
  - Provides linker files with the default installation
- Keil
  - Auto generated by project settings
  - Can be manually edited
- Simplicity IDE
  - Auto generated