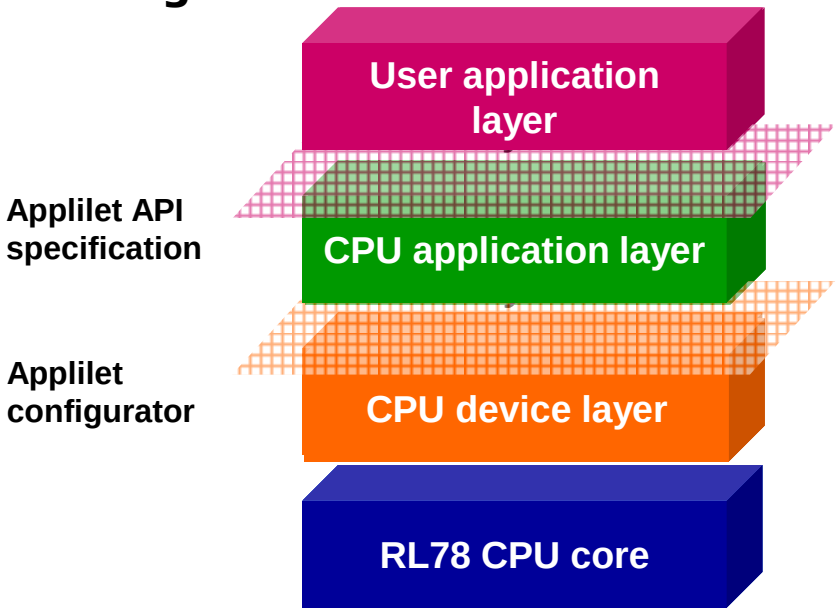
RL78/G13 Hands-On

Using e2studio and the integrated Auto Code Generator tool (known as Applilet) with RSK for RL78/G13

Renesas Electronics Europe

RL78 device driver code generator ('Applilet')

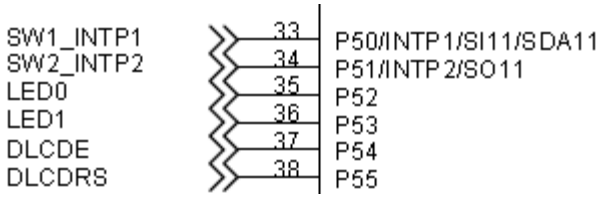
- **Free utility which generates device driver code to initialize and use RL78 on-chip peripherals**
- **Part of Project Generator in e2studio for RL78**
- **Easy to use graphical user interface (GUI)**
- **Common API for easy code porting across families**
- **Integrated project wizard guides user through the creation of a new project**
- **After peripheral configuration C source code can be generated**
- **Configuration changes automatically merged with existing user code**
  - User code in protected areas is saved during rebuild of the Applilet files



Simple Demo: Switch interrupt toggles an output

- Set up port pin P50 as an external interrupt INTP1
  - On the RSK this is connected to switch SW1
- Toggle LED0 in the INTP1 Interrupt Service Routine
  - LED0 is on port pin P52

From RSK RL78/G13 Schematic:



Simple Demo: Step-by-step Details (1)

- Start e2studio v2.1
  - Specify a new Workspace (e.g. C:\Workspace\RSKRL78G13), or choose an existing one

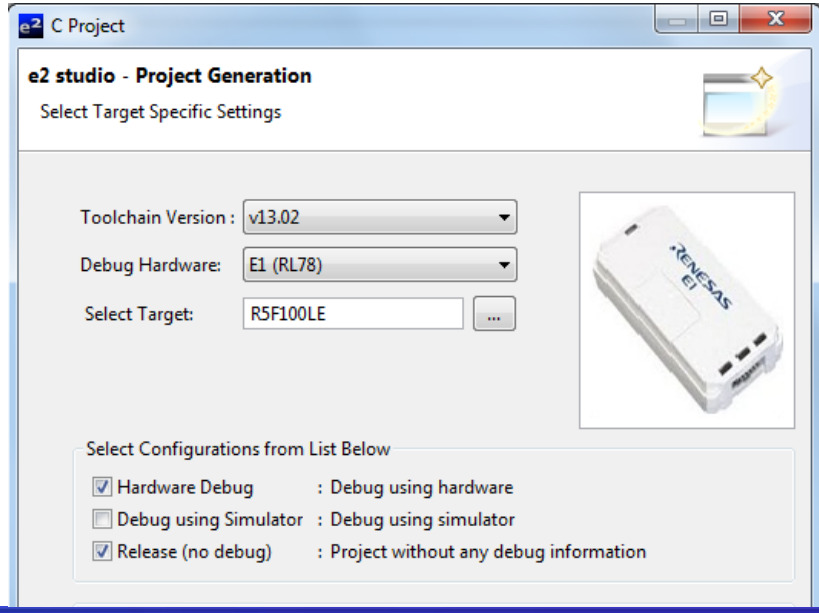
- Create a new C project
  - File > New > C Project
  - Choose a suitable project name, e.g. "INTP1_LED0"
  - Select Project type: 'Executable (Renesas) – Sample Project'
  - Toolchains: 'KPIT GNURL78-ELF Toolchain'

  Then click 'Next'

  Select
  - Toolchain version: 'v13.02'
  - Debug Hardware: 'E1 (RL78)'
  - Select Target:

  RL78 - G13 64pin 'R5F100LE'

  Then click 'Next'



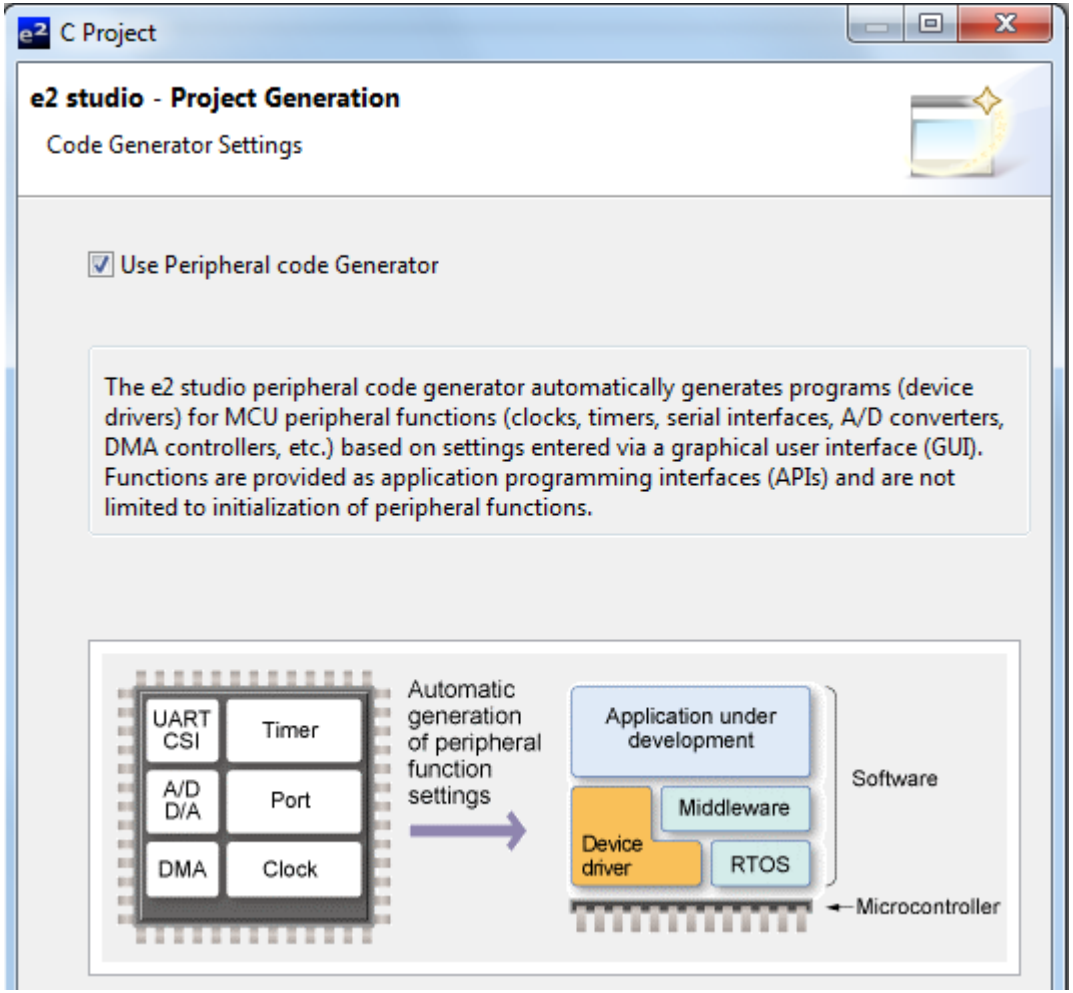
Simple Demo: Step-by-step Details (2)

- Select 'Use Peripheral Code Generator'

Then click 'Finish'

And 'Okay'

..



Simple Demo: Step-by-step Details (2)

- In Project Explorer view, open the Code Generator and Peripheral Functions folders, and double click on 'Clock Generator' Module
  - A separate window (Eclipse 'view') will open called 'Peripheral Functions'

- Configure 'Clock Generator' Module
  - In 'Pin Assignment' tab accept default and click 'Fix Settings'
  - Check 'Clock Setting' tab – if it looks okay, proceed
  - In 'On-chip debug setting' tab, click 'On-chip debug operation: Used'
  - Leave other tabs as they are

- Configure 'Port' Module
  - Double-click on 'Port' in Project Explorer view
  - Click on Port5 tab in Peripheral Functions view
  - Configure port P52 as an 'Out' (output pin) for LED0

Simple Demo: Step-by-step Details (3)

- Configure 'Interrupt' Module
  - Double-click on 'Interrupt' in Project Explorer view
  - Enable INTP1 interrupt for SW1
  - Leave as Valid Edge: 'Falling' and Priority 'Low'

- Disable Watchdog Timer
  - Double-click on 'Watchdog Timer' module
  - Select 'Unused' in Peripheral Functions view

- Generate Code
  - Click on the 'Generate Code icon  at the top-right of the Peripheral Functions view to generate the start-up and peripheral initialisation code for the project
  - This will now be available in the 'src' folder in the Project Explorer view

Simple Demo: Step-by-step Details (4)

- Add code to enable the external interrupt INTP1
  - In source file 'r_cg_intc.c' you will find the function to enable interrupt:

```
void R_INTC1_Start(void)
```

[Note: The function to initialise the interrupt - `void R_INTC_Create(void)` – is called automatically in the start-up code, in 'r_hardware_setup.c']

Open 'r_main.c' and add code to call this function in `void R_MAIN_UserInit(void)` function, directly before `EI();` statement:

```
R_INTC1_Start();
```

- Add code to toggle LED0
  - Open file 'r_cg_intc_user.c'
  - In ISR for INTP1 `void r_intc1_interrupt(void)`, insert code to toggle LED0 on port P52

e.g. `P5 ^= 0x04;`

[This statement will perform an Exclusive OR operation on Port 5 data register with 0b00000100 to toggle bit 2]

Simple Demo: Step-by-step Details (5)

- Build & Debug the project

  - Click on the Build icon in the toolbar to compile and link the code

  When you have eliminated any build errors and built the project successfully:

  - Connect the E1 debugger to your PC; Connect the E1 to the RSK
  - Click on the Debug icon in the toolbar to connect to the RSK

  [Note: The Debug Configuration should already have been generated automatically for the E1 and target device]

  The e2studio 'Perspective' will change to 'Debug'.

  When connection has been established and the code downloaded, the cursor should be on the PowerON Reset instruction.

  - Click the Resume icon twice to run the program

  (a breakpoint is automatically inserted at 'Main()'..)

  - Press SW1 switch to check operation of the code..

  [Note: as there is no debounce code, toggling is erratic)

Renesas Electronics Europe