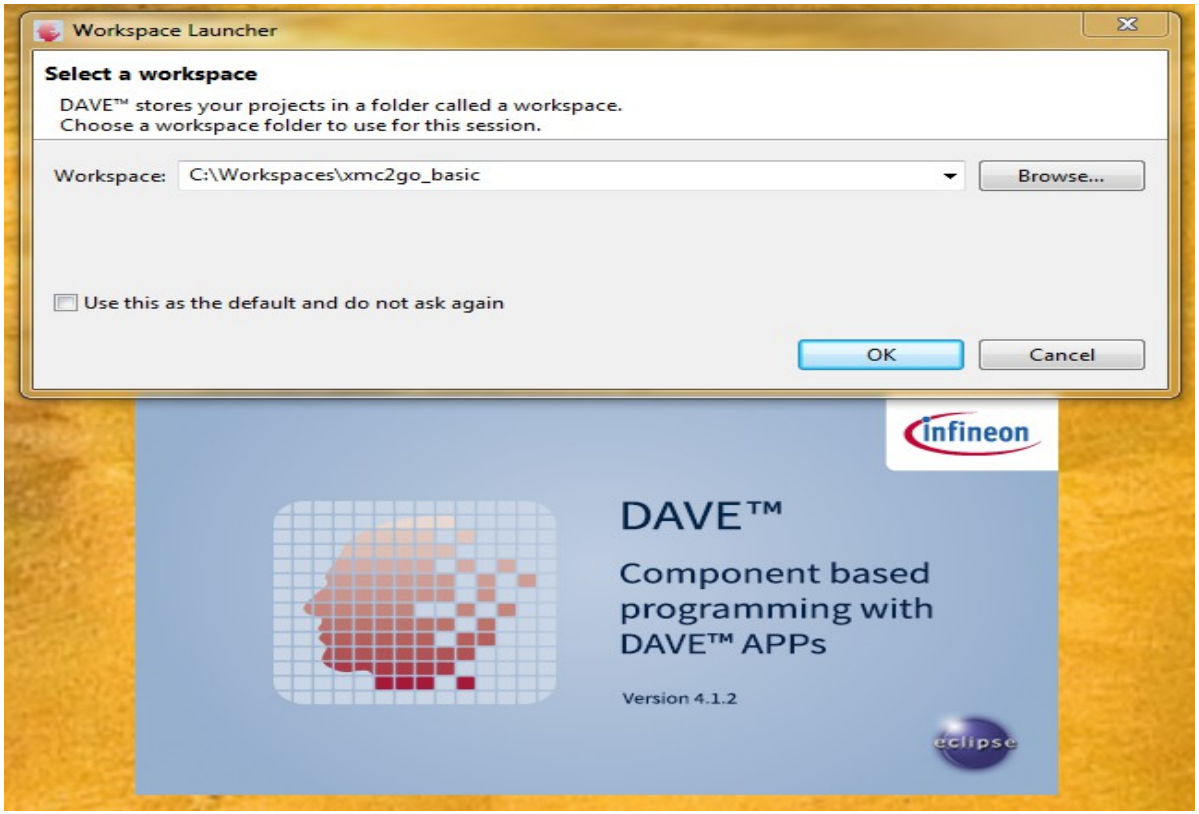
XMC2GO Labs

* xmc2go_basic_lab1 : Software/GPIO blink of two (2) LEDs
* xmc2go_basic_lab2 : Hardware/PWM blink of (2) LEDs
* xmc2go_basic_lab3 : UART and USB Host Communication
* xmc2go_basic_lab4 : SysTick Timer Interrupts
* xm2go_basic_lab5 : Typical System Configuration with ADC
* xmc2go_rtxrtos_lab1 : Four (4) RTX RTOS threads with two(2) UARTs
* xmc2go_freertos_lab1 : Two (2) Tasks with Send and Receive Queues

Source: PatternAgents

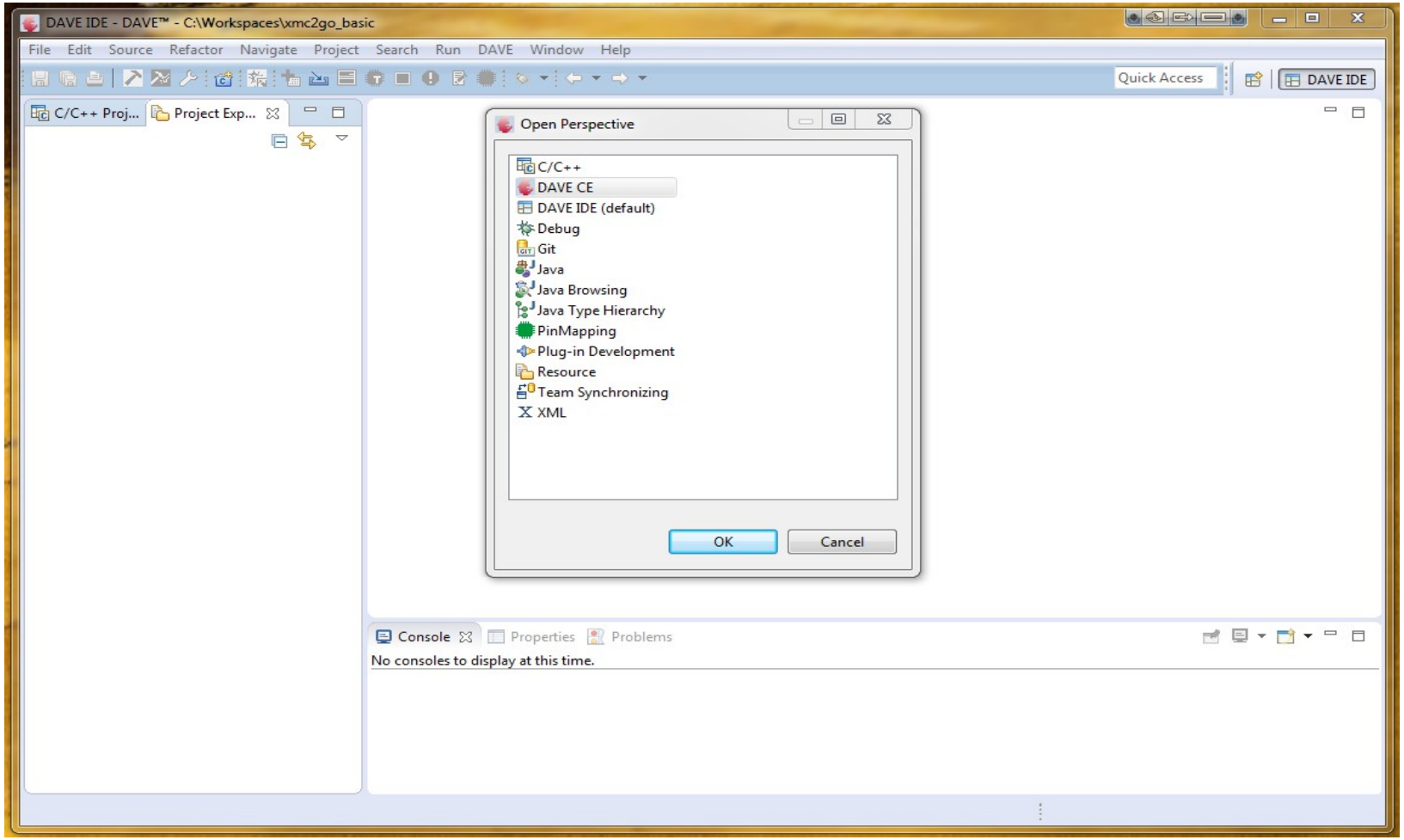
Start the DAVE IDE

* All Programs → DAVE-4.1.2 → DAVE_4.1.2
* Set the workspace to C:\Workspaces\xmc2go_basic



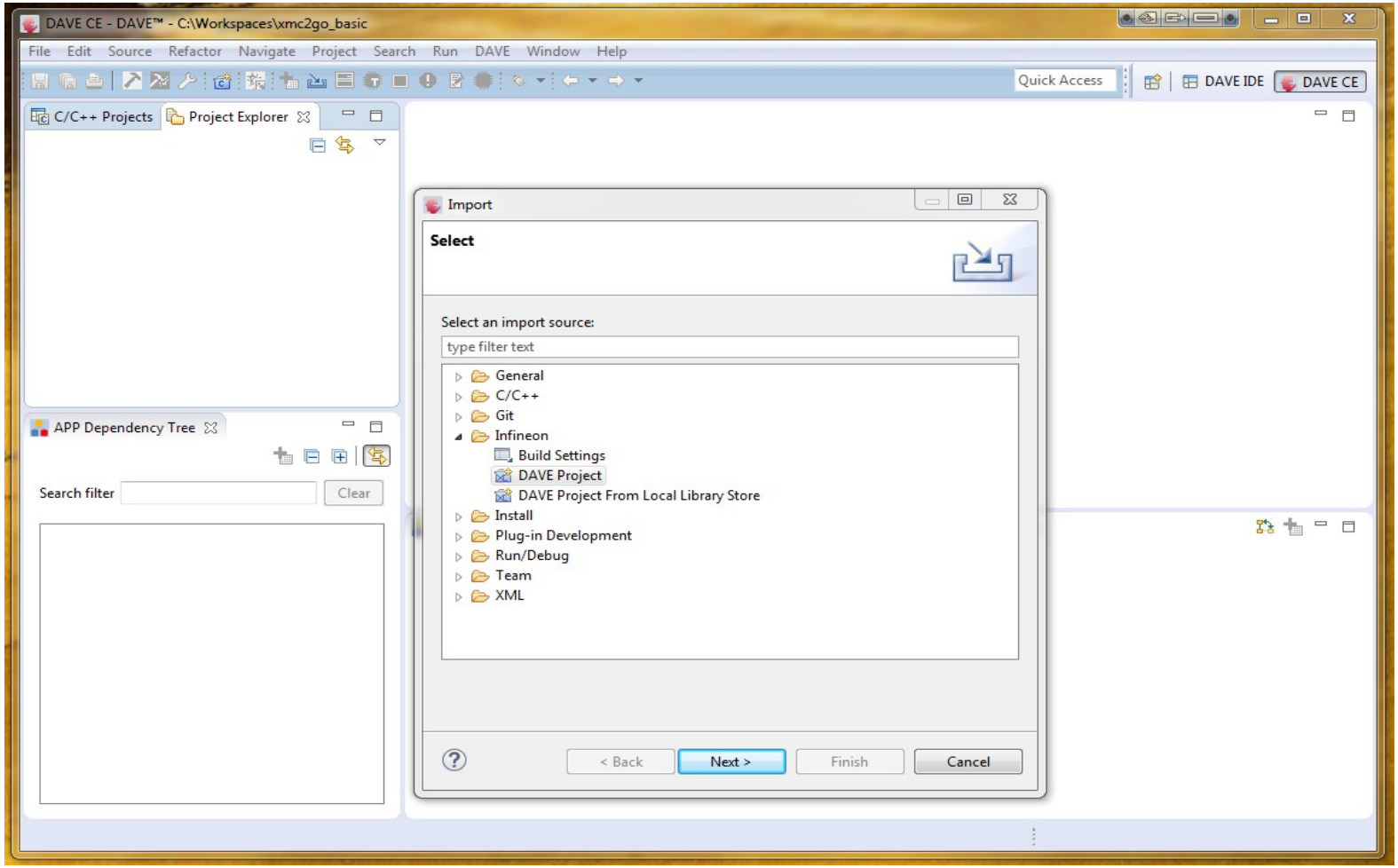
Source: PatternAgents

Open a DAVE CE Perspective



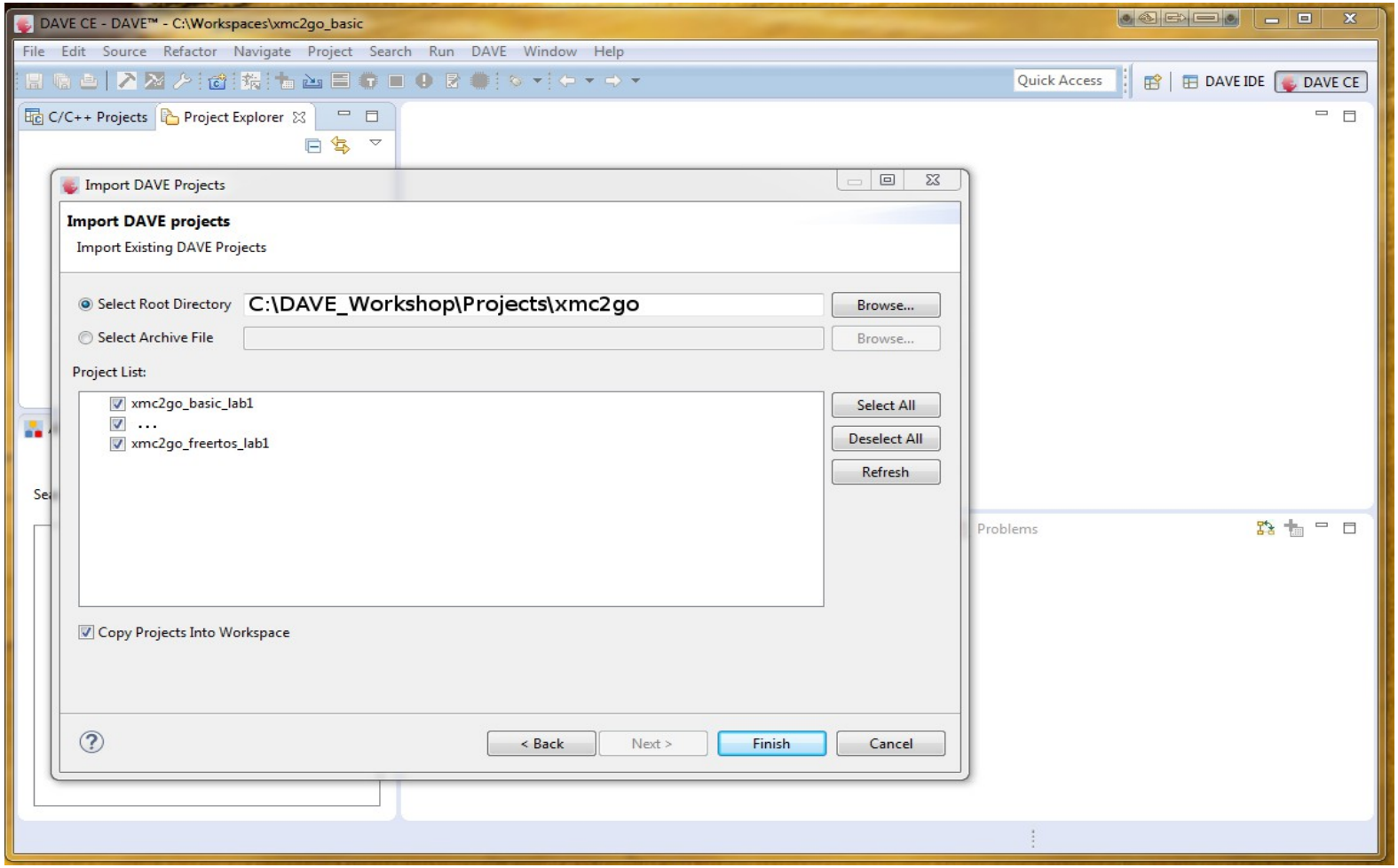
Source: PatternAgents

File → Import → DAVE project



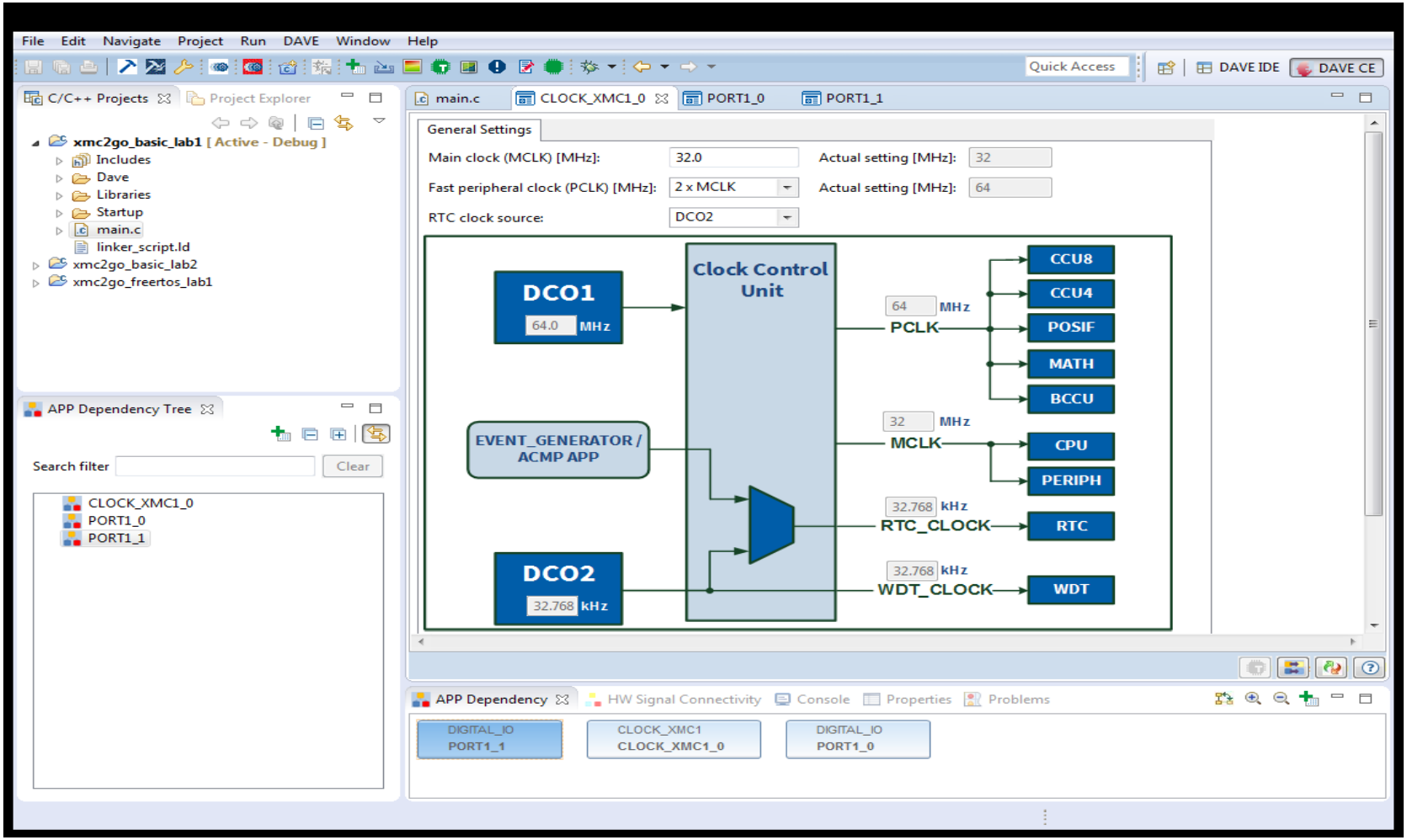
Source: PatternAgents

File → Import (Select Copy)



Source: PatternAgents

xmc2go_basic_lab1



Source: PatternAgents

main.c template

```c
#include <DAVE.h>

int main(void)
{
  DAVE_STATUS_t status;

  /** Initialize the DAVE APP run-time and return status  */
  status = DAVE_Init();

  if(status == DAVE_STATUS_FAILURE)
  {
    /* Placeholder for error handler code. */
    /* The while loop below can be replaced with a user defined error handler. */
    XMC_DEBUG("DAVE APPs initialization failed\n");

    while(1U)
    {
            /* XMC_DEBUG("Dave's not here, man...\n"); */
    }
  }

  /* Placeholder for user application code. */
  /* The while loop below can be replaced with user application code. */
  while(1U)
  {

  }
}
```

Source: PatternAgents

main.c xmc2go_basic_lab1

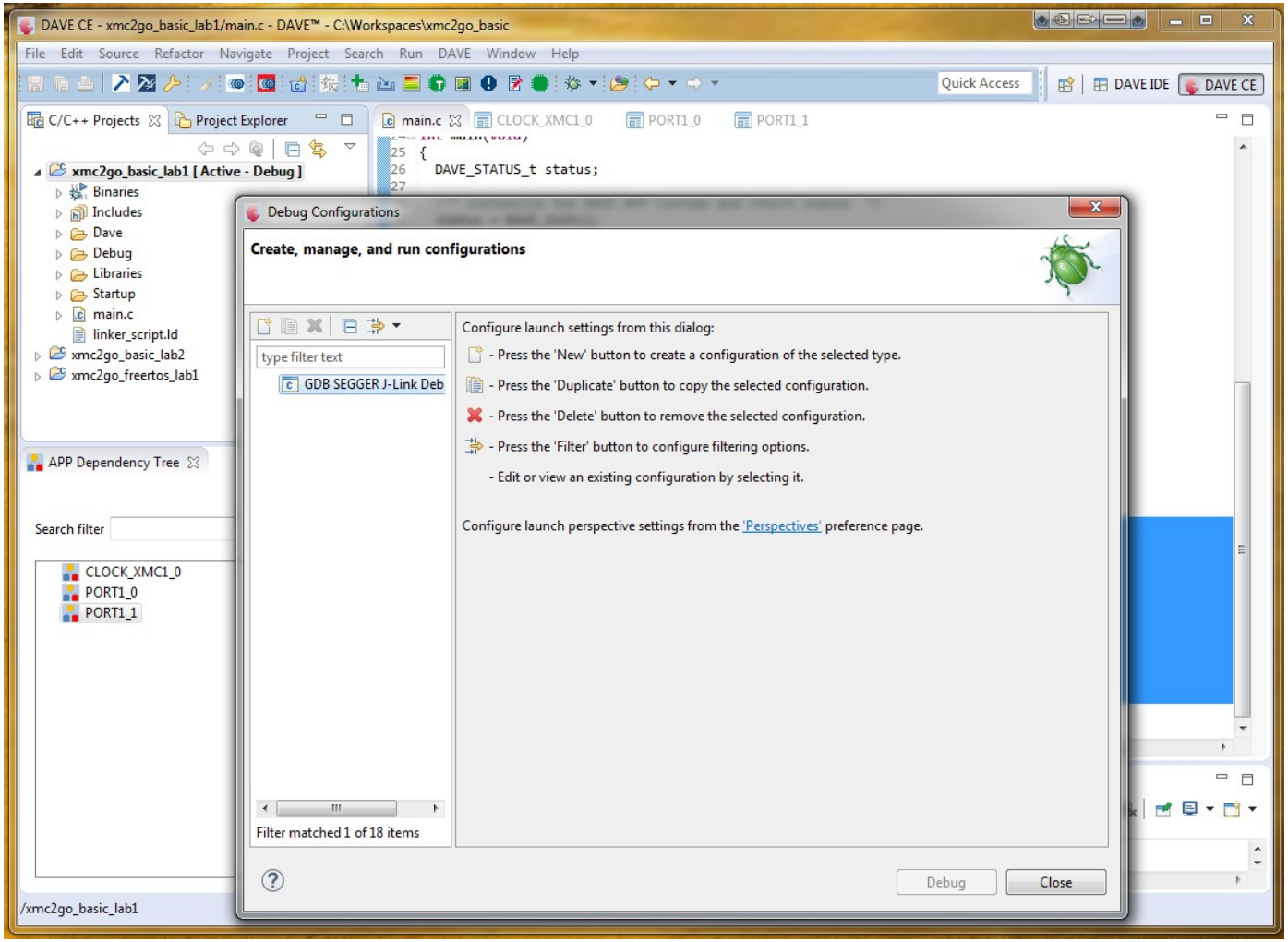
```
while(1U)
{

  long maxcount = 40000;
  long count;

  for(count = 0; count < maxcount; count++) {
             DIGITAL_IO_SetOutputHigh(&PORT1_0);
             DIGITAL_IO_SetOutputLow(&PORT1_1);
  }
  for(count = 0; count < maxcount; count++) {
             DIGITAL_IO_SetOutputLow(&PORT1_0);
             DIGITAL_IO_SetOutputHigh(&PORT1_1);
  }
}
}
```

Creates a simple back and forth blinking of the two LEDs

1) Generate Code (Runs the DAVE Hardware Solver)
2) Build Active Project (Runs the C/C++ Compiler and Linker)

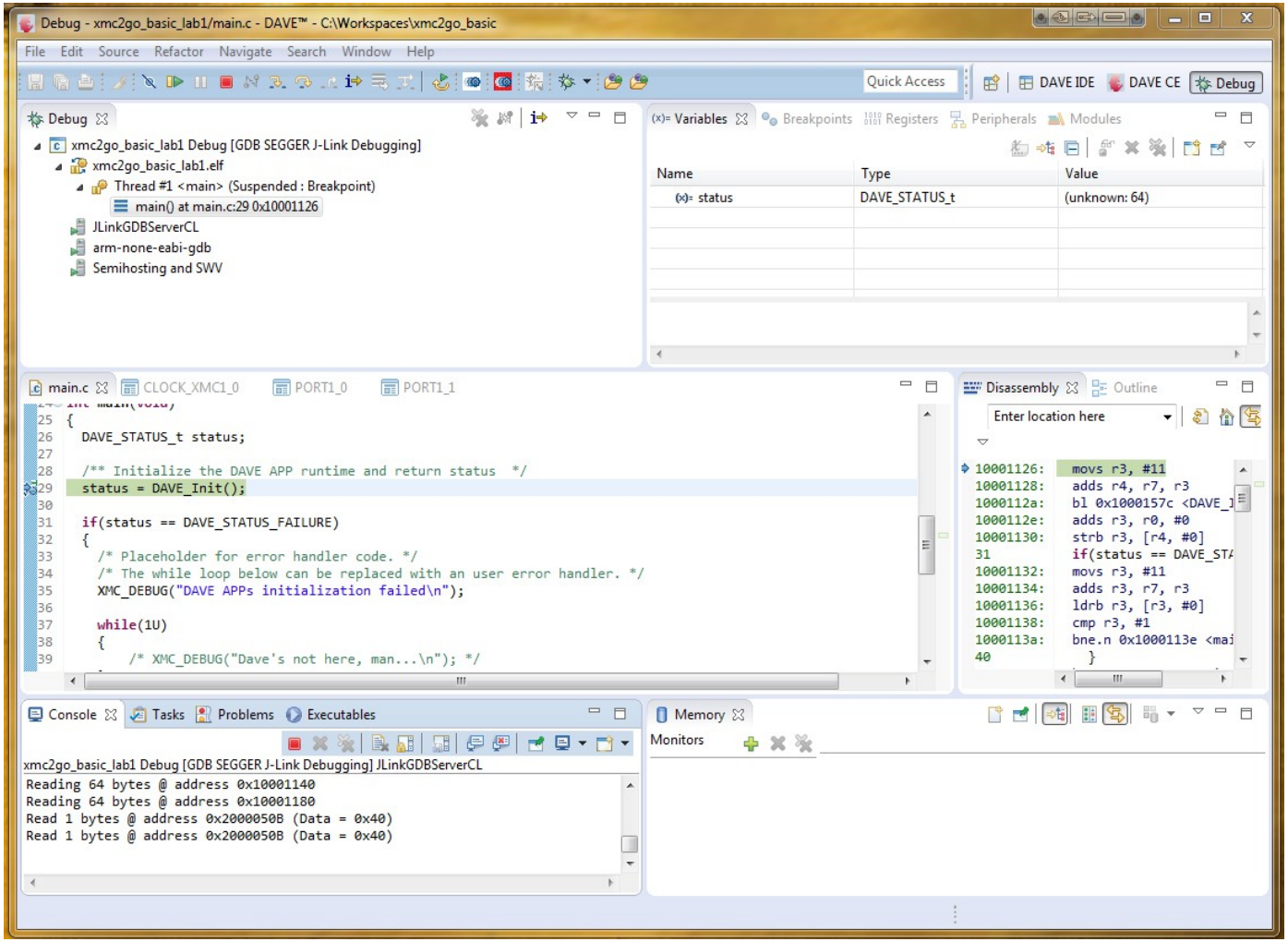
Source: PatternAgents

Debug : Configuration



Source: PatternAgents

Debug : Run/Step



Source: PatternAgents

xmc2go_basic_lab2

* xmc2go_basic_lab2 : Hardware/PWM blink of (2) LEDs
* “Hardware” based solution, starts automatically

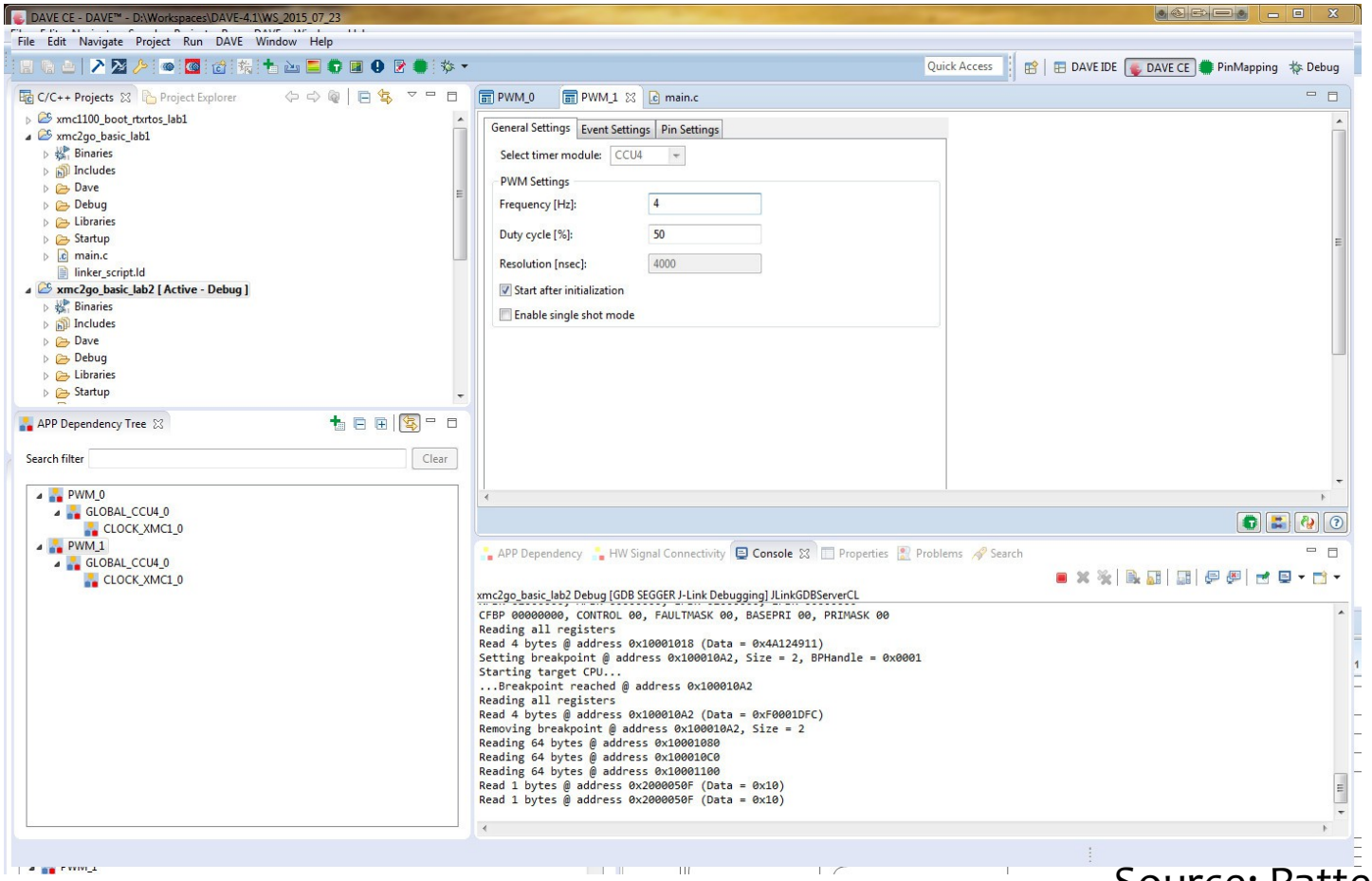
* PWM_0 is set to blink LED1 at a 1 Hz rate
* PWM_1 is set to blink LED2 at a 2Hz rate

* Build the project and enter the Debug Perspective
* Resume (F8) program execution and observe LEDs

Source: PatternAgents

xmc2go_basic_lab2

* Using the PWM_1 APP GUI, change the rate to 4 Hz



Source: PatternAgents

xmc2go_basic_lab3

* We will add two (2) UARTs to the previous lab
* UART_0 will be connected to the USB CDC/Debugger
* UART_0 will use P2.1 (TXD0) and P2.2 (RXD0)

* UART_1 will be available to use for a target device
* UART_1 will use P0.6 (TXD1) and P0.7 (RXD1)

* Both UARTS are set to 9600 baud, N, 8 , 1
  (open a terminal emulator to connect)

Source: PatternAgents

xmc2go_basic_lab4

* SysTick and Timer Interrupts
  (how to register a callback for timer interrupts)

```
void MySysTimer_Handler(void) {
            /* Toggle PORT1_0 every time we are called */
            DIGITAL_IO_ToggleOutput(&LED1);
}

/* init the  SYSTIMER (i.e. SysTick) */
  SYSTIMER_STATUS_t SysTimer_status;
  SysTimer_status = (SYSTIMER_STATUS_t)SYSTIMER_Init(&SYSTIMER_0); // Initialization of SYSTIMER APP
  if (SysTimer_status != SYSTIMER_STATUS_SUCCESS) {
              XMC_DEBUG("DAVE SYSTIMER Initialization Failed!\n");
  }

  /* create a software timer instance */
  uint32_t SysTimerId;
  SysTimerId = (uint32_t)SYSTIMER_CreateTimer(1000000U, SYSTIMER_MODE_PERIODIC, (void*)MySysTimer_Handler,
NULL);
  if (SysTimerId == 0U) {
              XMC_DEBUG("DAVE SYSTIMER CreateTimer Initialization Failed\n");
  }

  /* start the software timer instance */
  SysTimer_status = SYSTIMER_StartTimer(SysTimerId);
  if (SysTimer_status != SYSTIMER_STATUS_SUCCESS) {
              XMC_DEBUG("DAVE SYSTIMER StartTimer failed\n");
  }
```

Source: PatternAgents

xmc2go_basic_lab5

* Lab5 : Putting it all together to make a usable system
* We already have GPIO, PWM's, Systick Timer, and dual UART channels wired up and working
* Now add ADC_Measurement with six (6) Channels
* Read out ADC measurement to the UART/CDC interface and display on the terminal emulator

Source: PatternAgents

xmc2go_rtxrtos_lab1

* Add ARM RTS RTOS and ARM DSP APP to project
* Gives you the basis for starting real projects
* Adds Floating Point printf/scanf functionality (just to give you an idea of full function sizes)

* Project creates four (4) “threads” of execution

```
void thread_task_1(void const *args) {
  while (1) {
    DIGITAL_IO_ToggleOutput(&DIGITAL_IO_1);
    osDelay(500);
  }
}
osThreadDef(thread_task_1, osPriorityNormal, 1, 0);
```

Source: PatternAgents

xmc2go_rtxrtos_lab1

```
/*----------------------------------------------------*/
/* start up the ARM RTX RTOS                          */
/* NOTE: Increase the default number of threads...    */
/*       if you want more than 4 (default)            */
/*       If you experience problems using "printf",etc.*/
/*       then increase the default thread stack size   */
/*       i.e. (200 -> 512 bytes)                      */
/*----------------------------------------------------*/
osThreadCreate(osThread(thread_task_1), NULL);
osThreadCreate(osThread(thread_task_2), NULL);
osThreadCreate(osThread(thread_task_3), NULL);
osThreadCreate(osThread(thread_task_4), NULL);
osKernelStart();
/*----------------------------------------------------*/
/* the RTX RTOS takes over now, main doesn't return... */
/*----------------------------------------------------*/
while (1) {
  osDelay(1000);
}
```

* Build the project and start the debug perspective
* Threads 1,2,3 each toiggle an I/O output (LEDs)
* Thread 4 will output messages to the UART/CDC

Source: PatternAgents

Hand's On Lab Time

* Suggestions for other labs?
* What Applications (i.e. components) do you need ?
* Do the Applications (i.e. components) meet your specs?
* Select the XMC4500 in a new project to see more Apps (tip: Start with a larger model to see more Apps...)

Source: PatternAgents

More PA Workshops Coming

Please join the PatternAgents.com mailing lists to get early notice :

* Motor & Motion Control Workshop
  (Steppers, H-Bridge, and Brush-less Motor Control)

* Analog Design Workshop
  (Low Noise Analog and Mixed-Signal design)

* Blue tooth Low Energy Workshop
  (BTLE and Cloud integration and design)

* Other Topics (please give us feedback…)

Source: PatternAgents