XMC 2Go –
Write your first program Guide

Version 1.1

Preamble

Dear valued customer,

thank you for choosing the XMC 2Go Kit with the XMC1100 Microcontroller. Did you enjoy your first experience with the XMC 2Go already? And did you get appetite for more? Well, the following content is to give you an introduction how to write your first own program. Have fun!

Contents

› **Toolchain – import the factory default program into DAVE™**

› Modify the User LED blinking frequency

› Define your own messages send by XMC 2Go
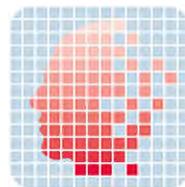
Import the factory default program into DAVE™

›   The easiest way to write an own program is to take an existing one and to modify it. To do so this guide will use the preprogrammed XMC_2Go_Initial_Start factory default program as reference.

›   Please have DAVE™ and the J-Link Driver installed as described in the "XMC 2Go – Initial start-up Guide", this is the prerequisite for the following steps.

›   Please download the XMC_2Go_Initial_Start project from the XMC 2Go product website (@ www.infineon.com/xmc-dev >> XMC™ 2Go >> Documents)
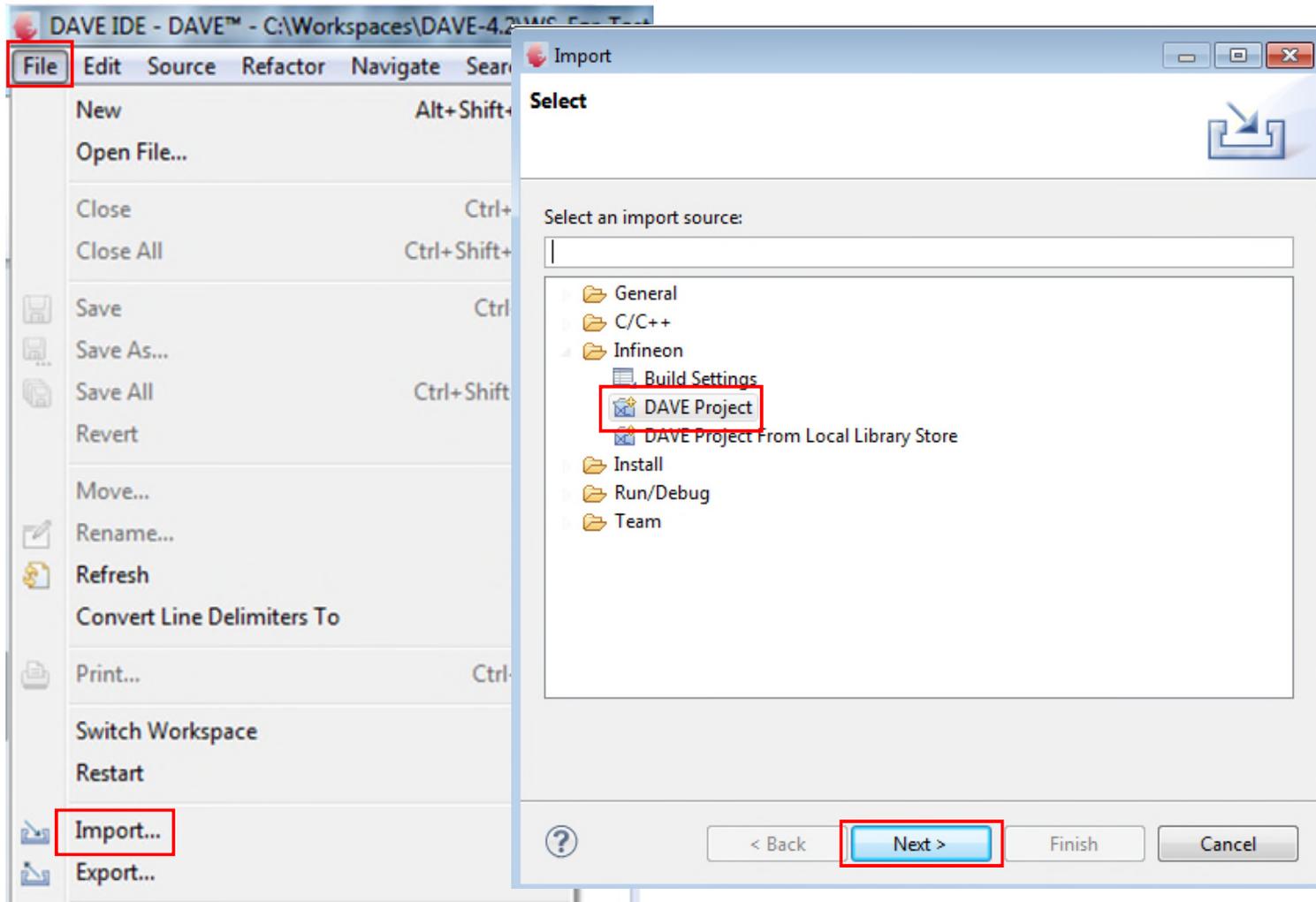
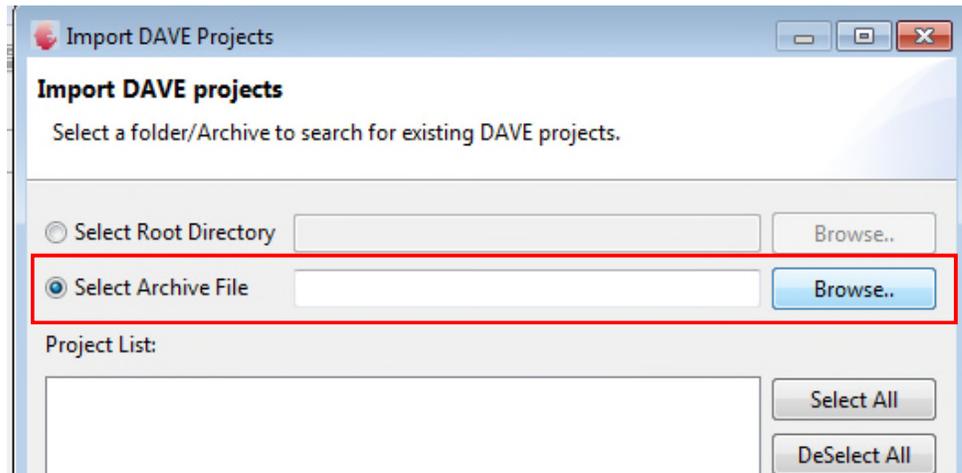ZIP   XMC™ 2Go Initial Start   › EN

›   Now launch DAVE ™

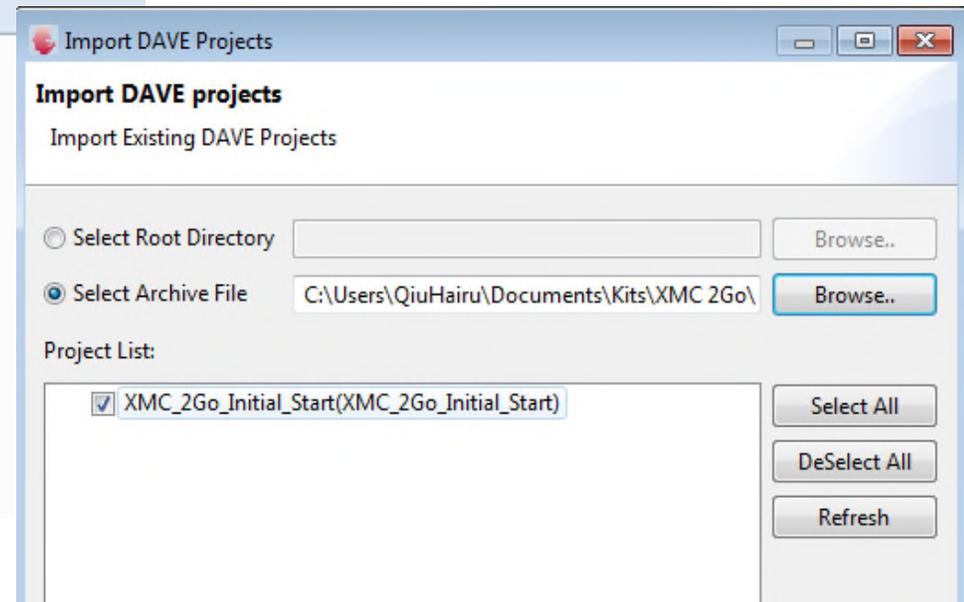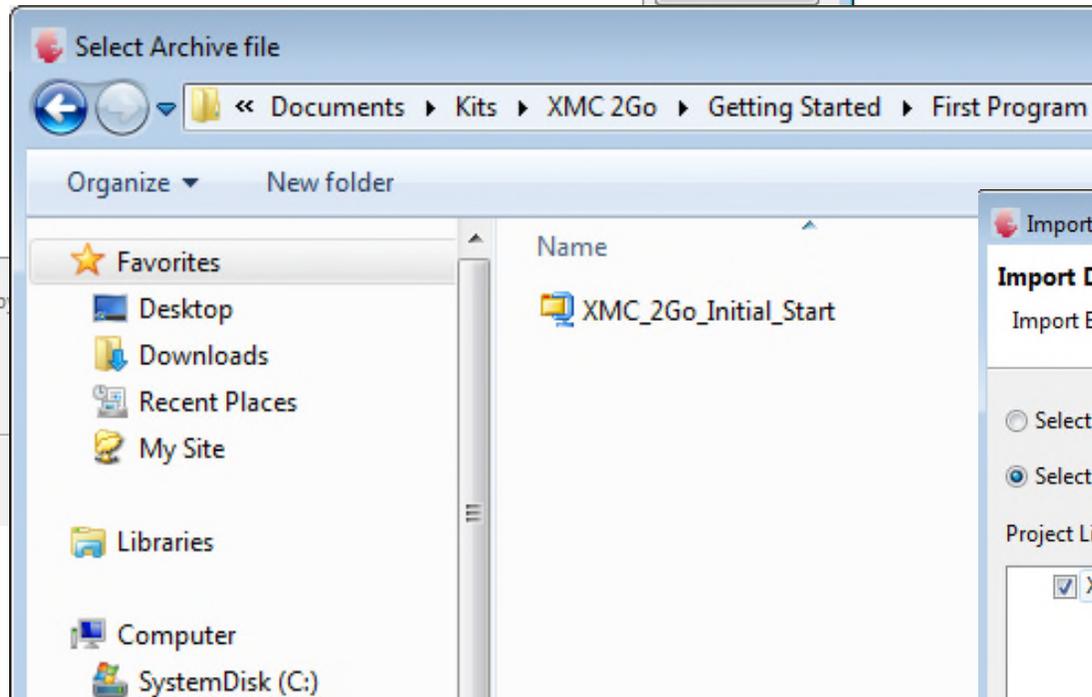# Import the factory default program into DAVE™ - cont'd

› Go to File, Import, Infineon, DAVE Project

Import the factory default program into DAVE™ - cont'd

› Browse to your local directory where XMC_2Go_Initial_Start is stored, click Open and Finish (Note: file names and project names could have been changed)

Import the factory default program into DAVE™ - cont'd

› Now the XMC_2Go_Initial_Start_v1.3 project is imported into DAVE™

› All relevant information is commented in main.c. Just go to collapsed view to get to the main.c file.

Contents

› Toolchain – import the factory default program into DAVE™

› **Modify the User LED blinking frequency**

› Define your own messages send by XMC 2Go

Modify the User LED blinking frequency

› The XMC_2Go_Initial_Start program toggle the User LEDs every 200 ms. The time base to do the toggling is derived from the system timer where a "timer tick" is generated every 100 ms. Hence 2 timer ticks are used setup the 200 ms toggle period.

› To change the blinking frequency of the User LEDs means nothing but to change the time period for toggling. This can be easily done by changing the number of timer ticks used to do the LED toggling in the main.c file.

› Following illustration shows the toggle period increase from 200ms to 400ms.

```
MERCHANTABILITY AND FITNESS FOR A PARTICUL
* INFINEON SHALL NOT, IN ANY CIRCUMSTANCES,
* OR CONSEQUENTIAL DAMAGES, FOR ANY REASON W
********************************************

#include <string.h>

#include "XMC1100.h"
#include "GPIO.h"

// Ticks are generated every 100ms
#define TICKS_PER_SECOND 10UL
// Update LED status every 200ms
#define TICKS_LED_EVENT 2UL
// Update message 2s
#define TICKS_UART_EVENT 20UL
```

```
-- consequence arising, for any s
********************************************

#include <string.h>

#include "XMC1100.h"
#include "GPIO.h"

// Ticks are generated every 100ms
#define TICKS_PER_SECOND 10UL
// Update LED status every 400ms
#define TICKS_LED_EVENT 4UL
// Update message 2s
#define TICKS_UART_EVENT 20UL
```
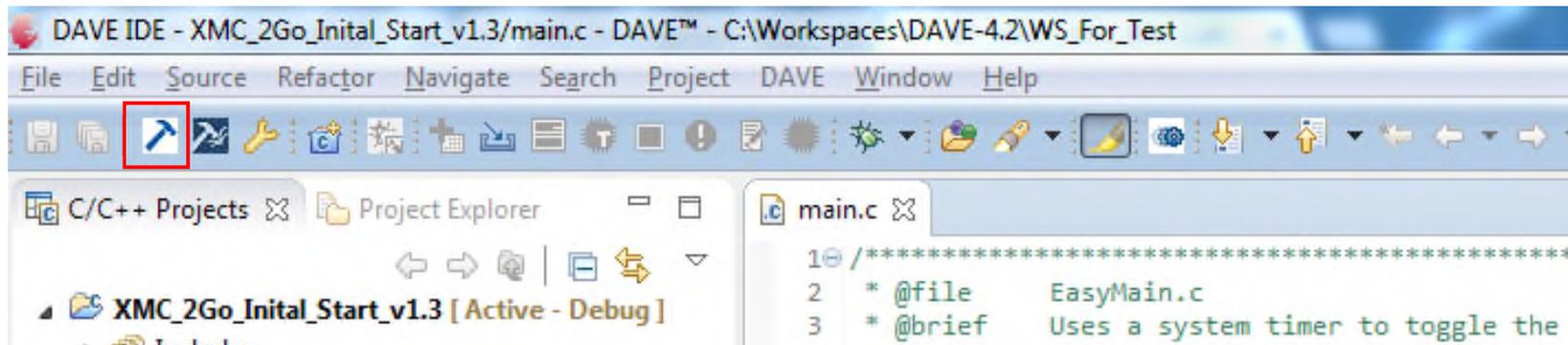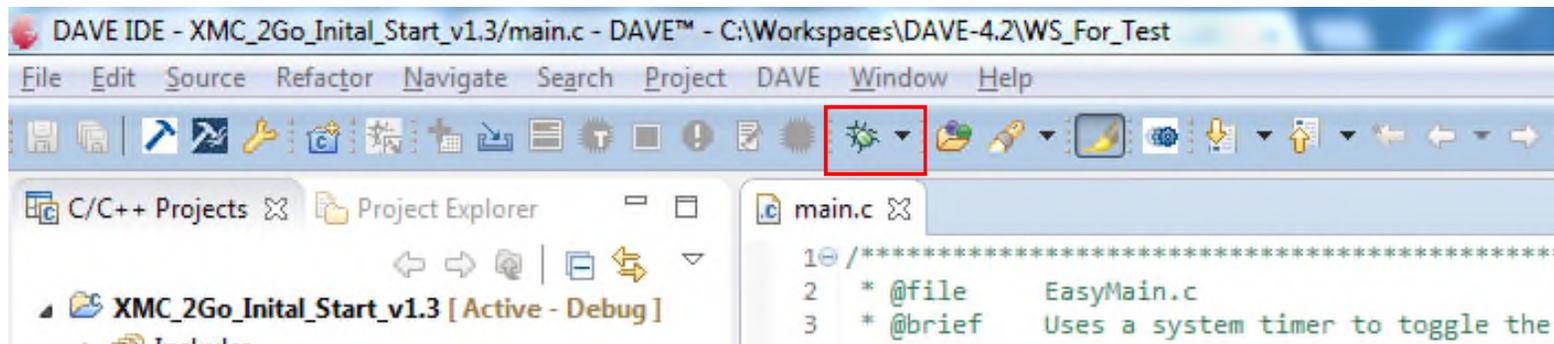
Modify the User LED blinking frequency – cont'd

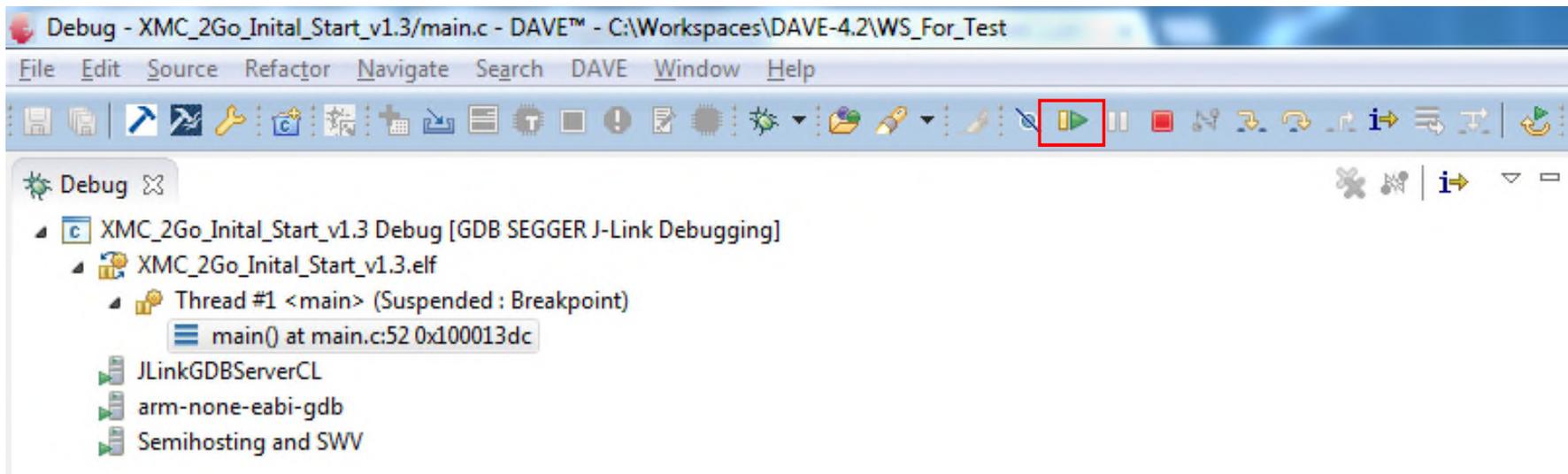› Don't forget to build & compile

› Go to debug for code download

Modify the User LED blinking frequency – cont'd

› Switch to Debug Perspective and hit the run button (Resume). The modified program is running now, watch the change of the User LED blinking frequency.


› Application Hint:

The XMC_2Go_Initial_Start sends every 2s messages via UART. Use the same principles as just described to change their send frequency. Just modify the "TICKS_UART_EVENT" value.

Contents

› Toolchain – import the factory default program into DAVE™

› Modify the User LED blinking frequency

› **Define your own messages send by XMC 2Go**

Define your own messages send by XMC 2Go

› The XMC_2Go_Initial_Start factory default program sends two predefined messages every 2s in an alternating way. These two messages can be received and displayed by any terminal program.

› The best way to show the messages on a terminal program is a recap of the "XMC 2Go – Initial start-up Guide", using the HTerm as terminal program.
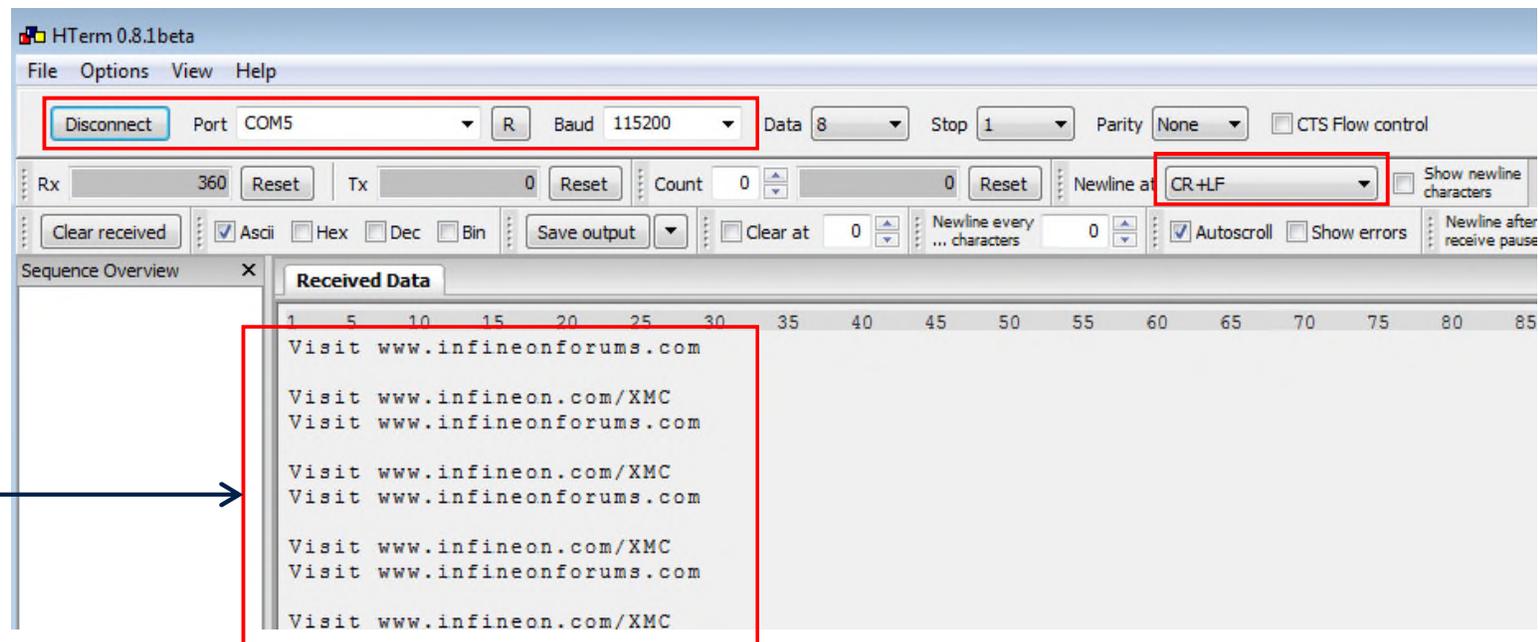
Predefined messages sent by XMC 2Go →

Define your own messages send by XMC 2Go – cont'd

› To define your own messages simply change the character string in the main.c file.

› Following illustration shows the message change from

   Visit www.infineon.com/XMC   to   *Wazzup*

   Visit www.infineonforums.com   *XMC 2Go rocks!*

```c
void SysTick_Handler(void)
{
    static uint32_t ticks_uart = 0;
    static uint32_t ticks_led = 0;
    static uint8_t message = 0;
    static const char *messages[] = {
            "Visit www.infineon.com/XMC\r\n",
            "Visit www.infineonforums.com\r\n\r\n"
    };

    uint8_t i;

    // Update LED status
    ticks_led++;
    if (ticks_led == TICKS_LED_EVENT) {
        P1_0_toggle();
        P1_1_toggle();
        ticks_led = 0;
    }

    // Update UART message
    ticks_uart++;
    if (ticks_uart == TICKS_UART_EVENT) {
```

```c
void SysTick_Handler(void)
{
    static uint32_t ticks_uart = 0;
    static uint32_t ticks_led = 0;
    static uint8_t message = 0;
    static const char *messages[] = {
            "Wazzup\r\n",
            "XMC 2Go rocks!\r\n\r\n"
    };

    uint8_t i;

    // Update LED status
    ticks_led++;
    if (ticks_led == TICKS_LED_EVENT) {
        P1_0_toggle();
        P1_1_toggle();
        ticks_led = 0;
    }

    // Update UART message
    ticks_uart++;
    if (ticks_uart == TICKS_UART_EVENT) {
        for(i = 0; i < strlen(messages[message]); i++) {
```
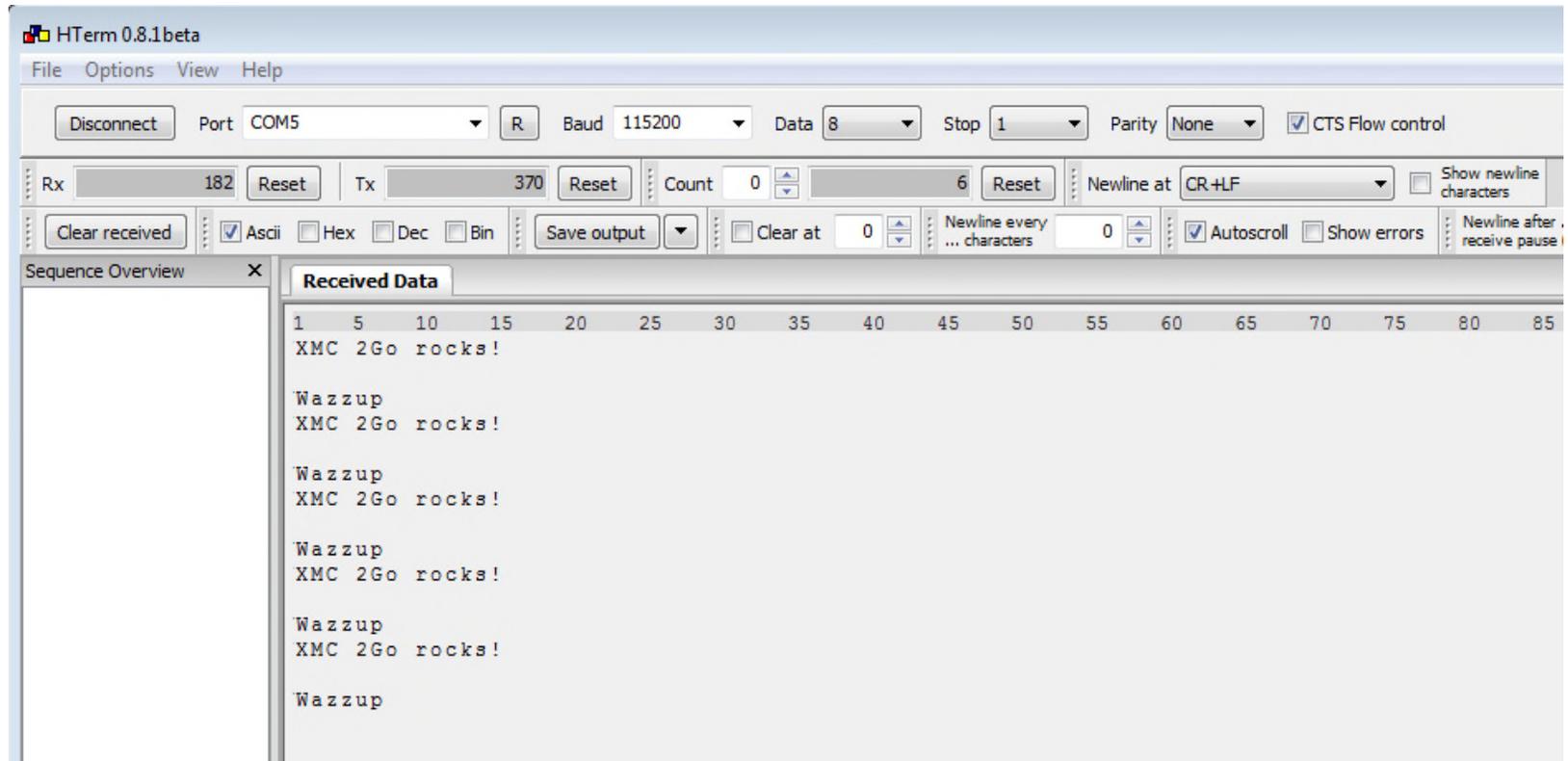
Define your own messages send by XMC 2Go – cont'd

› Build & compile, download and run the modified program as mentioned in the previous section.

› Don't forget to setup your terminal program with the respective UART settings as illustrated (recap the "XMC 2Go – Initial start-up Guide") and watch your own defined messages send by the XMC 2Go.

Part of your life. Part of tomorrow.

infineon