Developing Jade Robot BSP Instructions

Summary:

This document explains the process of creating a custom BSP for the Mimetics robot using the instructions provided by Martin Látal of Freescale as a base. This process starts with the BSP (Board Support Package) provided by Freescale for the twrk60d100m Tower development PCB and converts it into a BSP for the Freescale MK20DN512VLL10 used in the robot. Note that along with the BSP, the PSP, MFS, RTCS, Shell, USB Device and USB Device libraries as well as the CodeWarrior (CW) programming information for the device using the P&E Multilink-Universal device.

It is assumed that the instructions outlined here can be applied for different situations (ie target processor and source BSP) although it hasn't been tried.

This document was written for CodeWarrior 11.0 and MQX version 4.0.1

Target and Source Selection:

The MK20DN512VLL10 was chosen for the robot for the following features:

- IO including:
  - GPIO DI/DO
  - ADC Inputs
  - PWM
  - I^2C
  - I^2S with DAC Output
  - USB
  - UARTs
- 512 kBytes of Flash
- 128 kBytes of SRAM
- 100 MHz Maximum operating speed (96 MHz to be used in the product for USB operation)
- JTAG based programming/debug capability
- Flash Image update through Kinetis "Swap" feature
- LPM Mode rather than a switch

The PK60DN512VMD10 used on the twrk60d100m Tower development PCB has the following characteristics compared to the MK20DN512VLL10:

- Same IO Capabilities
- 512 kBytes of Flash
- 128 kBytes of Flash
- 100 MHz operating speed/(twrk40d100m is used for clocking option)
- Built in OSBDM programming/debugger

The PK60DN512VMD10 has the following capabilities/features that are not available on the MK20DN512VLL10:

- Ethernet Port
- ESDHC (high speed SD Card support)

These features are taken out of the custom bsp

NOTE: the PK60DN512VMD10 used on the twrk60d100m has a 60 MHz crystal and not an 8 MHz crystal.

The twrk40d100m bsp is used for clocking.

The twrk21d50m bsp is used for the lwadc code.

BSP Development Steps:

Note: the Martin Látal build steps were taken from: https://community.freescale.com/docs/DOC-94248

The contents of this web page has been added to the bottom of this document.

Basic j20 BSP Create from twrk60d100m:

0. If MQX/CodeWarrior are installed on the PC:
   - Uninstall MQX (first) and unpin from Taskbar
   - Uninstall CodeWarrior (Second)
   - Delete "RobotWorkspace2", "CW MCUv110.0" and "Freescale_MQX_4_0" in "C:\Freescale" folders if any present
   - Shutdown and restart PC

1. Install CodeWarrior 11.0 & MQX with proper setup:
   - Install CodeWarrior 11.0 using "Setup.exe" from "CW for MCU 11.0" in "C:\Users\User\Dropbox\Jade Robot Design Files Restored\Firmware"
   - From "C:\Freescale\CW MCU v11.0\eclipse" – "Pin to Taskbar" "cwide.exe"
   - Install MQX 4.0.1 by running "FSLMQXOS_4_0_1" from "C:\Users\User\Dropbox\Jade Robot Design Files Restored\Firmware"

2. Startup CodeWarrior:
   - Create "C:\Freescale\robotWorkspace2"
   - Install "Check for Updates" from "Help" dropdown. No "New Software" to install.
   - Restart CodeWarrior after installing the updates.

3. From "Import" and then "Other", "Import Working Sets":
   - twrk60d100m.wsd from C:\Freescale\Freescale_MQX_4_0\config\twrk60d100m\cw10
   - twrk40d100m.wsd from C:\Freescale\Freescale_MQX_4_0\config\twrk40d100m\cw10

4. In "dispatch.S" in the "PSP_Cortex" folder of the psp_twrk(40|60)d100m projects (same file), comment out lines as shown starting at 837 shown below to eliminate "Warning" during build (the "strexbeq" instruction is available in CodeWarrior 11):

```
//#if !defined(__CODEWARRIOR__)
        strexbeq r3, r2, [r0]       /* yes, try lock */
//#else
///* beloved CW10 dont know strexb */
// ASM_CONST16(0xe8c0)
// ASM_CONST16(0x2f43)
//#endif
```

5. Under the "Project" drop Down do:
   - "Clean" ("Clean all projects" but do not "Start build immediately")
   - "Build All" (There should NOT be any issues)

6. Shut down CodeWarrior.

7. Execute Step 1. of Martin Látal build steps adding the following line to psp_cpudef.h after line 71 in C:\Freescale\Freescale_MQX_4_0\mqx\source\psp\cortex_m:
```
#define PSP_CPU_MK20DN512
(PSP_CPU_NUM(PSP_CPU_ARCH_ARM_CORTEX_M4,
PSP_CPU_GROUP_KINETIS_K2X, 7))
```

8. Execute Step 2. of Martin Látal build steps by copying MK20D10.h from C:\Freescale\CW MCU v11.0\MCU\ProcessorExpert\lib\Kinetis\iofiles into C:\Freescale\Freescale_MQX_4_0\mqx\source\psp\cortex_m\cpu

9. Execute Step 3. of Martin Látal build steps by adding the following lines to kinetis.h after line 49 in C:\Freescale\Freescale_MQX_4_0\mqx\source\psp\cortex_m:
```
#elif   (MQX_CPU == PSP_CPU_MK20DN512)
 #include "MK20D10.h"
```

10. Execute Step 4. of Martin Látal build steps by running the BSPCloningWizard found in C:\Freescale\Freescale_MQX_4_0\tools\BSPCloningWizard. Latest BSP is called "j20" and use "twrk60d100m" as the base.
    - **NOTE:** May need 32bit version of Java JRE installed before BSPCloningWizard works
    - Run "Generate MQX Projects"
    - Close BSPCloningWizard

11. Execute Step 5. of Martin Látal build steps. Edit user_config.h found in C:\Freescale\Freescale_MQX_4_0\config\j20
    - Change the MQX_CPU in to PSP_CPU_MK20DN512
    - Delete/comment out the line: `#define BSPCFG_ENABLE_ESDHC 1`

12. Steps 6-12 of Martin Látal build steps were ignored because there are no changes to the IO of the BSP is required and the output of the BSPCloningWizard is not correct and changes must be made to get a working BSP that can be programmed into an MK20DN512VLL10.

13. Edit j20.mem in C:\Freescale\Freescale_MQX_4_0\mqx\source\bsp\j20\cw\dbg to have a "usederivative" of "MK20D10" instead of "MK60D10".

14. Edit bsp_prv.h in C:\Freescale\Freescale_MQX_4_0\mqx\source\bsp\j20:
    - Use "#ifdef DOINJ20" to take out the line:
```
extern const ESDHC_INIT_STRUCT _bsp_esdhc0_init;
```

15. Edit psp_j20.bat in C:\Freescale\Freescale_MQX_4_0\mqx\build\bat to use MK20D10.h instead of the MK60D10.h

16. Edit psp_j20.sh in C:\Freescale\Freescale_MQX_4_0\mqx\build\bat to use MK20D10.h instead of the MK60D10.h

17. Edit bsp_j20.bat in C:\Freescale\Freescale_MQX_4_0\mqx\build\bat:
    - Delete all the lines starting with:

```
copy %MQXROOTDIR%\mqx\source\io\enet\
copy %MQXROOTDIR%\mqx\source\io\esdhc
copy %MQXROOTDIR%\mqx\source\io\sdcard
```

    - Replace all instances of "k60" with "k20"

18. Edit bsp_j20.sh in C:\Freescale\Freescale_MQX_4_0\mqx\build\bat:
    - Delete all the lines starting with:

```
cp -f "${MQXROOTDIR}/mqx/source/io/enet
cp -f "${MQXROOTDIR}/mqx/source/io/esdhc
cp -f "${MQXROOTDIR}/mqx/source/io/sdcard
```

    - Replace all instances of "k60" with "k20"

19. Edit ProcessorExpert.pe in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\bsp_j20:
    - Change all instances of "K60" to "K20"

20. Edit j20.h in C:\Freescale\Freescale_MQX_4_0\mqx\source\bsp\j20:
    - Use "#ifdef DOINJ20" to take out the section:

```
/*--------------------------------------------------------------------
--------------
**                          ESDHC
*/
```

    - Use "#ifdef DOINJ20" to take out the section:

```
/*--------------------------------------------------------------------
--------------
**                          Ethernet Info
*/
```

    - Use "#ifdef DOINJ20" to take out the section:

```
/*
** ESDHC device
** MGCT: <option type="bool"/>
*/
```

21. Copy mqx_TWRK60D100M.xml as mqx_j20.xml in C:\Freescale\CW MCU v11.0\MCU\lib\wizard_data\mqx\4.0\arm

22. Edit mqx_j20.xml in C:\Freescale\CW MCU v11.0\MCU\lib\wizard_data\mqx\4.0\arm:
    - Replace every instance of twrk60d100m with j20
    - Replace every instance of K60DN512M10 with K20DN512M10
    - Replace every instance of K60 with K20

23. Create the "j20" folder in "C:\Freescale\Freescale_MQX_4_0\build"
    - Copy make folder and contents from C:\Freescale\Freescale_MQX_4_0\build\twrk60d100m into C:\Freescale\Freescale_MQX_4_0\build\j20
    - Edit each of the Makefile files in the different folders of C:\Freescale\Freescale_MQX_4_0\build\j20\make and change every instance of twrk60d100m to j20

24. Edit .project in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\bsp_j20:
    - Add New Lines after ever "</link>" to make the file easier to read
    - Replace every instances of "k60" to "k20".
    - Delete all "<link>…</link>" for 'enet'
    - Delete all "<link>…</link>" for 'esdhc'
    - Delete all "<link>…</link>" for 'macnet'
    - Delete all "<link>…</link>" for 'ksz8041'
    - Delete all "<link>…</link>" for 'sdcard'

25. Edit .cproject in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\bsp_j20:
    - Delete all "<listOptionValue…/>" for 'enet'
    - Delete all "<listOptionValue…/>" for 'esdhc'
    - Delete all "<listOptionValue…/>" for 'macnet'
    - Delete all "<listOptionValue…/>" for 'ksz8041'
    - Delete all "<listOptionValue…/>" for 'sdcard'

26. Edit .project in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\psp_j20:
    - Add New Lines after ever "</link>" to make the file easier to read
    - Replace every instance of "MK60D10.h" with "MK20D10.h"
    - Replace every instances of "k60" to "k20".
    - Delete all "<link>…</link>" for 'enet'
    - Delete all "<link>…</link>" for 'esdhc'
    - Delete all "<link>…</link>" for 'macnet'
    - Delete all "<link>…</link>" for 'ksz8041'
    - Delete all "<link>…</link>" for 'sdcard'

27. Edit .cproject in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\psp_j20:
    - Delete all "<listOptionValue…/>" for 'enet'
    - Delete all "<listOptionValue…/>" for 'esdhc'
    - Delete all "<listOptionValue…/>" for 'macnet'
    - Delete all "<listOptionValue…/>" for 'ksz8041'
    - Delete all "<listOptionValue…/>" for 'sdcard'

28. Edit j20.yml in C:\Freescale\Freescale_MQX_4_0\tools\generator\records:
    - Replace every instance of "MK60D10.h" with "MK20D10.h"
    - Replace every instance of K60 with K20
    - Replace every instance of k60 with k20
    - **NOTE:** Leaving in "MK60DN512.mem" for now

29. Edit adc_mk20.c in the C:\Freescale\Freescale_MQX_4_0\mqx\source\io\adc\kadc:
    - Add the lines after the "#elif (MQX_CPU = PSP_CPU_MK20DX256)" block:

```
#elif (MQX_CPU == PSP_CPU_MK20DN512)
   static const pointer adc_address[] = {
            (pointer)ADC0_BASE_PTR,
            (pointer)ADC1_BASE_PTR
};
```

30. Edit spi_mk20.c in the C:\Freescale\Freescale_MQX_4_0\mqx\source\io\spi:
    - Add the lines after the "#elif (MQX_CPU = PSP_CPU_MK20DX256)" block:

```
#elif (MQX_CPU == PSP_CPU_MK20DN512)
static const pointer dspi_address[] =
{
    (pointer)SPI0_BASE_PTR,
    (pointer)SPI1_BASE_PTR,
    (pointer)SPI2_BASE_PTR,
};

static const uint_32 /*PSP_INTERRUPT_TABLE_INDEX*/
dspi_vectors[][1] =
{
    { INT_SPI0 },
    { INT_SPI1 },
    { INT_SPI2 },
};
```

31. Edit i2c_mk20.c in C:\Freescale\Freescale_MQX_4_0\mqx\source\io\i2c:
    - After the "`#if (MQX_CPU == PSP_CPU_MK20DX256)`" block in the _bsp_get_i2c_base_address method add the block of conditional code:

```
#elif (MQX_CPU == PSP_CPU_MK20DN512)
      case 1:
         addr = (pointer) I2C1_BASE_PTR;
         break;
```

    - After the "`#elif (MQX_CPU == PSP_CPU_MK20DX256)`" block in the _bsp_get_i2c_vector method add the block of conditional code:

```
#elif (MQX_CPU == PSP_CPU_MK20DN512)
      case 1:
         vector = INT_I2C1;
         break;
```

32. In serl_mk20.c in C:\Freescale\Freescale_MQX_4_0\mqx\source\io\serial:
    - After the "`#elif (MQX_CPU == PSP_CPU_MK20DX256)`" block in the _bsp_get_serial_base_address method add the block of conditional code:

```
#elif (MQX_CPU == PSP_CPU_MK20DN512)
   case 3:
      addr = (pointer)UART3_BASE_PTR;
      break;
   case 4:
      addr = (pointer)UART4_BASE_PTR;
      break;
   case 5:
      addr = (pointer)UART5_BASE_PTR;
      break;
```

33. Startup CodeWarrior
    - Import j20.wsd library projects definition file from C:\Freescale\Freescale_MQX_4_0\config\j20\cw10

34. Edit bsp_cm.c from the j20 project "j20 BSP Files" and replace its contents with the code from bsp_cm.c from the twrk40d100m project "twrk40d100m BSP files" – This is to ensure the clocking is correct for the Jade PCB

35. Edit init_gpio.c in the bsp_j20\j20 BSP Files folder:
    - Use "#ifdef DOINJ20" to take out the method `_bsp_enet_io_init`.
    - Use "#ifdef DOINJ20" to take out the method `_bsp_esdhc_io_init`.

36. Edit bsp.h in the bsp_j20\j20 BSP Files folder:
    - Use "#ifdef DOINJ20" to take out the lines:

```
#include <enet.h>
#include <macnet_mk60.h>
```

    - Use "#ifdef DOINJ20" to take out the lines:

```
#include <esdhc.h>
#include <sdcard.h>
#include <sdcard_spi.h>
#include <sdcard_esdhc.h>
```

    - Change "#include <adc_mk60.h>" to "#include <adc_mk20.h>"
    - Change "#include <flash_mk60.h>" to "#include <flash_mk20.h>"
    - Use "#ifdef DOINJ20" to take out the lines:

```
_mqx_int _bsp_enet_io_init(_mqx_uint);
boolean  _bsp_get_mac_address(uint_32,uint_32,_enet_address);
```

    - Use "#ifdef DOINJ20" to take out the line:

```
extern const SDCARD_INIT_STRUCT _bsp_sdcard0_init;
```

37. Under the “Project” drop Down do:
    - “Clean” (“Clean all projects” but do not “Start build immediately”)
    - “Build All”
    - Should **NOT** have **ANY** problems (Warnings or Errors)
    - Remove the twrk60d100m and twrk40d100m bsp, psp and other projects (But do NOT remove from disk)

38. Create the “z1st_LED_DEMO” “debug” project (source files in document below) and test it on a Jade Robot using the j20 libraries:
    - **NOTE:** Use the Freescale Compiler, NOT GCC when building the project
    - Console “printf” operation (Probably won’t work)
    - LED (lwgpio) operation
    - Application must build and execute without issues before going on

39. Test “z1st_LED_DEMO” “release” project.
    - Change “z1st_LED_DEMO”, “bsp_j20” & “psp_j20” projects to “release”
    - Do a clean of each project and a build
    - Application should build & execute without issues (the same as before) before going on
    - Restore the “z1st_LED_DEMO”, “bsp_j20” & “psp_j20” projects to “debug”

Updated Interrupt I²C Driver Install:

40. Shut down CodeWarrior.

41. Copy i2c_int_k_fb.c from C:\Freescale\Files to Install for Robot\into bsp_ - Peripheral IO Drivers - i2c – int into C:\Freescale\Freescale_MQX_4_0\mqx\source\io\i2c\int

    - Here is i2c_int_k_fb.c file: 
      i2c_int_k_fb.c

42. Edit .project in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\bsp_j20 and add the following lines after the "<name>Peripheral IO Drivers/i2c/int/i2c_int.c</name>" Block on Line 1030:

```
      <link>
            <name>Peripheral IO
Drivers/i2c/int/i2c_int_k_fb.c</name>
            <type>1</type>
      <locationURI>MQX_ROOT_DIR/mqx/source/io/i2c/int/i2c_int_k_f
b.c</locationURI>
      </link>
```

43. Edit j20.h in C:\Freescale\Freescale_MQX_4_0\mqx\source\bsp\j20

    - Add the lines after the "** Interrupt-driven I2C1 device" block:

```
/*
** Enhanced Interrupt-driven I2C0 device
** MGCT: <option type="bool"/>
*/
#ifndef BSPCFG_ENABLE_II2C0_FB
    #define BSPCFG_ENABLE_II2C0_FB                0
#endif
/*
** Enhanced Interrupt-driven I2C1 device
** MGCT: <option type="bool"/>
*/
#ifndef BSPCFG_ENABLE_II2C1_FB
    #define BSPCFG_ENABLE_II2C1_FB                0
#endif
```

44. Edit j20.h in C:\Freescale\Freescale_MQX_4_0\mqx\source\bsp\j20 and delete the lines:

```
* Other Serial console options:(do not forget to enable
BSPCFG_ENABLE_TTY define if changed)
 *      "ittyf:"     OSJTAG-COM  interrupt mode
 *      "ttyd:"      TWR-SER     polled mode
 *      "ittyd:"     TWR-SER     interrupt mode
 *      "iodebug:"   IDE debug console
```

- **NOTE:** The reason for this command isn't obvious. I've put in "#ifdef DOINJ20" around the code block but the build failed because "BSP_DEFAULT_IO_CHANNEL" is undefined. **Left as is**

45. Edit user_config.h in C:\Freescale\Freescale_MQX_4_0\config\j20 and add the lines after the "#define BSPCFG_ENABLE_II2C1":
```
#define BSPCFG_ENABLE_II2C0_FB    0
#define BSPCFG_ENABLE_II2C1_FB    0
```

46. Restart CodeWarrior.

47. In i2c_ki2c.h found in CodeWarrior's bsp_j20\Peripheral IO Drivers\i2c project, in the "Function Prototypes" area, add the line:
```
extern uint_32 _ki2c_int_fb_install (char_ptr,
KI2C_INIT_STRUCT_CPTR);
```

48. In init_bsp.c found in CodeWarrior's bsp_j20\j20 BSP Files project, after the check for "BSPCFG_ENABLE_II2C1" add the lines:
```
#if BSPCFG_ENABLE_II2C0_FB
_ki2c_int_fb_install("ii2c0fb:", &_bsp_i2c0_init);
#endif
#if BSPCFG_ENABLE_II2C1_FB
_ki2c_int_fb_install("ii2c1fb:", &_bsp_i2c1_init);
#endif
```

49. Open user_config.h in CodeWarrior and set the I2C options as:
```
#define BSPCFG_ENABLE_I2C0        0
#define BSPCFG_ENABLE_I2C1        0
#define BSPCFG_ENABLE_II2C0       0
#define BSPCFG_ENABLE_II2C1       0
#define BSPCFG_ENABLE_II2C0_FB    1
#define BSPCFG_ENABLE_II2C1_FB    1
```

50. Under the "Project" drop Down do:
    - "Clean" ("Clean all projects" but do not "Start build immediately")
    - "Build All"/Should **NOT** have **ANY** problems (Warnings or Errors)

lwadc Driver Install:

51. Shut Down CodeWarrior

52. Copy lwadc_k20.h into C:\Freescale\Freescale_MQX_4_0\mqx\source\io\lwadc

53. Copy init_lwadc.c from C:\Freescale\Freescale_MQX_4_0\mqx\source\bsp\twrk21d50m into C:\Freescale\Freescale_MQX_4_0\mqx\source\bsp\j20

54. Edit lwadc_k20.c and add the lines after (pointer)ADC0_BASE_PTR with:
```
#if (MQX_CPU == PSP_CPU_MK20DX256)
    , (pointer)ADC1_BASE_PTR
#elif (MQX_CPU == PSP_CPU_MK20DN512)
    , (pointer)ADC1_BASE_PTR
#endif
```

55. Open .cproject for editing in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\bsp_twrk21d50m and in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\bsp_j20
    - Find each instance of "lwadc" in the .cproject from the bsp_twrk21d50m and copy them into the same position in the .cproject for bsp_j20 (there are four instances, at time of writing and they are all after "lwgpio" entries).
    - Close the two .projects editors, saving the modified version for bsp_j20.

56. Open .project for editing in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\bsp_twrk21d50m and in C:\Freescale\Freescale_MQX_4_0\mqx\build\cw10\bsp_j20
    - Copy the lwadc entries from the bsp_twrk21d50m into j20 in the same positions
    - Change the reference from lwadc_k21.h to lwadc_k20.h in j20
    - Change the "<name>twrk21d50m BSP Files/…" Reference and folders to "j20"
    - Close the two .projects editors, saving the modified version for bsp_j20.

57. Restart CW

58. Edit adc_mk20.h in bsp_j20\Peripheral IO Drivers\adc and add the lines after the "#define ADC_HAS_PGA 0" statement:
```
#define ADC_NUM_DEVICES 2
#define ADC_HW_CHANNELS 30
```

59. Edit init_lwadc.c in bsp_j20\j20 BSP Files and add the code:
```
#if (MQX_CPU == PSP_CPU_MK20DN512)
const LWADC_INIT_STRUCT lwadc1_init = {
        /* The number of ADC peripheral, use adc_t enum from
PSP */
        1,
```

```
            /* The clock source, selects the best from BUSCLK and
BUSCLK/2 */
            LWADC_CLK_BUSCLK_ANY,
            /* The clock divisor for ADC. use the fastest one */
            LWADC_DIV_ANY,
            /* ADC high speed control, see ADC_HSC enum */
            LWADC_HSC_NORMAL,
            /* ADC low power control, see ADC_LPC enum */
            LWADC_LPC_NORMAL,
            /* The calibration data pointer */
            NULL,
            /* ADC interrupt vector */
            INT_ADC1,
            /* ADC interrupt vector */
            BSP_ADC1_VECTOR_PRIORITY,

            BSP_ADC_VREF_DEFAULT
        };
#endif
```

60. Open bsp_prv.h in bsp_j20\j20 BSP Files and add the lines:
```
extern const LWADC_INIT_STRUCT lwadc0_init;
extern const LWADC_INIT_STRUCT lwadc1_init;
```

61. Open bsp.h in bsp_j20\j20 BSP Files and, after "#include <adc_kadc.h>" add the line:
```
#include <lwadc_kadc.h>
```

62. Open init_bsp.c in bsp_j20\j20 BSP Files and, after the "#if BSPCFG_ENABLE_ADC1" statement add the lines:
```
#if BSPCFG_ENABLE_LWADC0
        _lwadc_init(&lwadc0_init);
#endif
#if BSPCFG_ENABLE_LWADC1
        _lwadc_init(&lwadc1_init);
#endif
```

63. Edit user_config.h in C:\Freescale\Freescale_MQX_4_0\config\j20 and add the lines:
```
#define BSPCFG_ENABLE_LWADC0                    0
#define BSPCFG_ENABLE_LWADC1                    0
```

64. Edit j20.h in C:\Freescale\Freescale_MQX_4_0\mqx\source\bsp\j20
    - After the "ADC1" block, add the code:

```
/*
** LWADC0
** MGCT: <option type="bool"/>
*/
#ifndef BSPCFG_ENABLE_LWADC0
    #define BSPCFG_ENABLE_LWADC0                0
```

```
#endif
/*
** LWADC1
** MGCT: <option type="bool"/>
*/
#ifndef BSPCFG_ENABLE_LWADC1
    #define BSPCFG_ENABLE_LWADC1                0
#endif
```

- After the "**#define** BSP_PDB_VECTOR_PRIORITY (3)" statement, add the line:

```
#define BSP_ADC_VREF_DEFAULT             (3300)
```

65. Edit Makefile(s) in:
    - C:\Freescale\Freescale_MQX_4_0\build\j20\make\bsp & C:\Freescale\Freescale_MQX_4_0\build\twrk21d50m\make\bsp
    - C:\Freescale\Freescale_MQX_4_0\build\j20\make\psp & C:\Freescale\Freescale_MQX_4_0\build\twrk21d50m\make\psp
    - C:\Freescale\Freescale_MQX_4_0\build\j20\make\mfs & C:\Freescale\Freescale_MQX_4_0\build\twrk21d50m\make\mfs
    - C:\Freescale\Freescale_MQX_4_0\build\j20\make\rtcs & C:\Freescale\Freescale_MQX_4_0\build\twrk21d50m\make\rtcs
    - C:\Freescale\Freescale_MQX_4_0\build\j20\make\shell & C:\Freescale\Freescale_MQX_4_0\build\twrk21d50m\make\shell
    - C:\Freescale\Freescale_MQX_4_0\build\j20\make\usbd & C:\Freescale\Freescale_MQX_4_0\build\twrk21d50m\make\usbd
    - C:\Freescale\Freescale_MQX_4_0\build\j20\make\usbh & C:\Freescale\Freescale_MQX_4_0\build\twrk21d50m\make\usbh
        - Copy all instances of "lwadc" in twrk21d50m version into j20 version
        - Make sure that the twrk21d50m references are changed to j20
        - Make sure all "k60" references are changed to "k20" (just seems to be bsp)
        - Delete Code for "enet"
        - Delete Code for "esdhc"
        - Delete Code for "sdcard"

66. Start up CodeWarrior

67. Open "Properties" for each project (in "j20" and the others) and go to "C/C++ Build" ➔ "Settings" ➔ "Compiler" ➔ "Input". In "-i (Include User Search Paths (-i)" and after "lwgpio" add:
    `"${MQX_ROOT_DIR}/mqx/source/io/lwadc"`

68. Test the set up of the lwadc device driver by doing a clean and build of all the projects
    - Reset z1st_LED_DEMO

69. Edit init_gpio.c in bsp_j20\j20 BSP Files in the method "_bsp_i2c_io_init":
    - Move line "`#define ALT2 0x2`" to top of file

- After the "#define ALT2 0x2", add the line "#define ALT6 0x6"
- Change default I2C0 to PTB1 (SDA)/PTB0 (SCL) by changing the code for "case 0:" to:

```
pctl = (PORT_MemMapPtr)PORTB_BASE_PTR;

pctl->PCR[1] = PORT_PCR_MUX(ALT2) | PORT_PCR_ODE_MASK;
pctl->PCR[0] = PORT_PCR_MUX(ALT2) | PORT_PCR_ODE_MASK;
```

- Change default I2C1 to PTE0 (SDA)/PTE1 (SCL) changing the code for "case 1:" to:

```
pctl = (PORT_MemMapPtr)PORTE_BASE_PTR;

pctl->PCR[1] = PORT_PCR_MUX(ALT6) | PORT_PCR_ODE_MASK;
pctl->PCR[0] = PORT_PCR_MUX(ALT6) | PORT_PCR_ODE_MASK;
```

70. Under the "Project" drop Down do:
    - "Clean" ("Clean all projects" but do not "Start build immediately")
    - "Build All"/Should **NOT** have **ANY** problems (Warnings or Errors)

Low Power Operation Modifications for Mimetics Robot:

71. No CodeWarrior is Started for the changes below

72. In `user_config.h` change value of MQX_ENABLE_LOW_POWER to "1"

73. Edit init_lpm.c in j20 BSP Files:
    - Change the block following:

```
/* LPM_OPERATION_MODE_SLEEP */
```

to

```
/* LPM_OPERATION_MODE_SLEEP */
{
    LPM_CPU_POWER_MODE_WAIT, /* Index of predefined mode */
//      LPM_CPU_POWER_MODE_FLAG_SLEEP_ON_EXIT,  /* Additional
mode flags */
    0,    /* NO Additional mode flags */
    0,    /* Mode wake up events from pins 0..3 */
    0,    /* Mode wake up events from pins 4..7 */
    0,    /* Mode wake up events from pins 8..11 */
    0x80, /* Mode wake up events from pins 12..15 */
//  Modified by myke predko to wake up from D6 (Down Button)
    0     /* Mode wake up events from internal input sources */
},
```

    - Change the block following:

```
/* LPM_OPERATION_MODE_STOP */
```

to

```
/* LPM_OPERATION_MODE_STOP */
{
  LPM_CPU_POWER_MODE_LLS,  /* Index of predefined mode */
  0,     /* Additional mode flags */
  0,     /* Mode wake up events from pins 0..3 */
  0,     /* Mode wake up events from pins 4..7 */
  0,     /* Mode wake up events from pins 8..11 */
  0x80,  /* Mode wake up events from pins 12..15 */
  LLWU_ME_WUME0_MASK  /* Mode wake up events from internal input
sources - LPT */
}
```

74. Edit init_sci in j20 BSP Files:
    - Change the _bsp_sci0_operation_modes[LPM_OPERATION_MODES] = values to:

```
{
  /* LPM_OPERATION_MODE_RUN */
  {
    IO_PERIPHERAL_PIN_MUX_ENABLE | IO_PERIPHERAL_CLOCK_ENABLE
| IO_PERIPHERAL_MODULE_ENABLE,
    0,
    0,
    0
```

```
        },

        /* LPM_OPERATION_MODE_WAIT */
        {
            IO_PERIPHERAL_PIN_MUX_ENABLE | IO_PERIPHERAL_CLOCK_ENABLE
| IO_PERIPHERAL_MODULE_ENABLE,
            0,
            0,
            0
        },

        /* LPM_OPERATION_MODE_SLEEP */
        {
            IO_PERIPHERAL_PIN_MUX_ENABLE | IO_PERIPHERAL_CLOCK_ENABLE
| IO_PERIPHERAL_MODULE_ENABLE | IO_PERIPHERAL_WAKEUP_ENABLE |
IO_PERIPHERAL_WAKEUP_SLEEPONEXIT_DISABLE,
            0,
            0,
            0
        },

        /* LPM_OPERATION_MODE_STOP */
        {
            IO_PERIPHERAL_PIN_MUX_DISABLE |
IO_PERIPHERAL_CLOCK_DISABLE,
            0,
            0,
            0
        }
    };
```

- Change the _bsp_sci1_operation_modes[LPM_OPERATION_MODES] = values to:

```
    {
        /* LPM_OPERATION_MODE_RUN */
        {
            IO_PERIPHERAL_PIN_MUX_ENABLE | IO_PERIPHERAL_CLOCK_ENABLE
| IO_PERIPHERAL_MODULE_ENABLE,
            0,
            0,
            0
        },

        /* LPM_OPERATION_MODE_WAIT */
        {
            IO_PERIPHERAL_PIN_MUX_ENABLE | IO_PERIPHERAL_CLOCK_ENABLE
| IO_PERIPHERAL_MODULE_ENABLE,
            0,
            0,
            0
        },
```

```
        /* LPM_OPERATION_MODE_SLEEP */
        {
            IO_PERIPHERAL_PIN_MUX_ENABLE | IO_PERIPHERAL_CLOCK_ENABLE
| IO_PERIPHERAL_MODULE_ENABLE | IO_PERIPHERAL_WAKEUP_ENABLE |
IO_PERIPHERAL_WAKEUP_SLEEPONEXIT_DISABLE,
            0,
            0,
            0
        },

        /* LPM_OPERATION_MODE_STOP */
        {
            IO_PERIPHERAL_PIN_MUX_DISABLE |
IO_PERIPHERAL_CLOCK_DISABLE,
            0,
            0,
            0
        }
    };
```

75. Under the “Project” drop Down do:
    - “Clean” (“Clean all projects” but do not “Start build immediately”)
    - “Build All”/Should **NOT** have **ANY** problems (Warnings or Errors)

76. Close CodeWarrior

Flashx Modifications for "Swap" Operation:

77. Copy "swapsupport" folder from swapsupport.zip into C:\Freescale\Files to Install for Robot:

swapsupport.zip

78. Replace "flashx.c", "flashx.h" and "flashprv.h" in C:\Freescale\Freescale_MQX_4_0\mqx\source\io\flashx from swapsupport

79. Replace "flash_ftfl.c" and "flash_ftfl_prv.h" in C:\Freescale\Freescale_MQX_4_0\mqx\source\io\flashx\freescale from swapsupport

80. Start up Code Warrior

81. Under the "Project" drop Down do:
    - "Clean" ("Clean all projects" but do not "Start build immediately")
    - "Build All"/Should **NOT** have **ANY** problems (Warnings or Errors)

BSP Modifications for Mimetics Robot:

82. Change default UART0 (for Bluetooth Module) to PTA14 (TX)/PTA15 (RX) by editing init_gpio.c in bsp_j20\j20 BSP Files in the method "_bsp_serial_io_init" and changing the code for "case 0:" to:

```
pctl = (PORT_MemMapPtr)PORTA_BASE_PTR;
if (flags & IO_PERIPHERAL_PIN_MUX_ENABLE)
{
    /* PTA15 as RX function (Alt.3) + drive strength */
    pctl->PCR[15] = 0 | PORT_PCR_MUX(3) | PORT_PCR_DSE_MASK;
    /* PTA14 as TX function (Alt.3) + drive strength */
    pctl->PCR[14] = 0 | PORT_PCR_MUX(3) | PORT_PCR_DSE_MASK;
}
if (flags & IO_PERIPHERAL_PIN_MUX_DISABLE)
{
    /* PTA15 default */
    pctl->PCR[15] = 0;
    /* PTA14 default */
    pctl->PCR[14] = 0;
}
```

NOTE: There is no need to modify UART1 (Camera UART – TTYB) In init_gpio.c because the default is the same as in the product.

83. In j20.h found in bsp_j20\j20 BSP Files change "`#define BSP_ALARM_FREQUENCEY 200`" to "`#define BSP_ALARM_FREQUENCEY 500`"

84. Under the "Project" drop Down do:
    - "Clean" ("Clean all projects" but do not "Start build immediately")
    - "Build All"/Should **NOT** have **ANY** problems (Warnings or Errors)

BSP Modifications for High-Speed Camera Operation:

85. Add the following lines to serial.h in bsp_j20\Peripheral IO Drivers\serial at the end of the "Serial I/O IOCTL commands":

```
//  Extensions for High-Speed Camera Operation
#define IO_IOCTL_SERIAL_JPEG_IN1_OUT0     _IO(IO_TYPE_SERIAL,0x1B)
#define IO_IOCTL_SERIAL_JPEG_DONE_POLL    _IO(IO_TYPE_SERIAL,0x1C)
#define IO_IOCTL_SERIAL_GET_IN0_CMD       _IO(IO_TYPE_SERIAL,0x1D)
#define IO_IOCTL_SERIAL_GET_IN0_BYTES     _IO(IO_TYPE_SERIAL,0x1E)
#define IO_IOCTL_SERIAL_JPEG_ROVER        _IO(IO_TYPE_SERIAL,0x1F)
```

86. Replace serl_int_kuart.c in bsp_j20\Peripheral IO Drivers\serial\polled with the inserted:


serl_int_kuart.c

87. Edit serl_kuart.h in bsp_j20\Peripheral IO Drivers\serial: and add the following line at the end of the kuart_info_struct typdef:

```
uint_32     IMAGECOMPLETE_FLAG;
```

88. Replace serl_pol_kuart.c in bsp_j20\Peripheral IO Drivers\serial\polled with the inserted:


serl_pol_kuart.c

89. Under the "Project" drop Down do:
    - "Clean" ("Clean all projects" but do not "Start build immediately")
    - "Build All"/Should **NOT** have **ANY** problems (Warnings or Errors)

New Application Notes:

1. Open "Properties" for project and go to "C/C++ Build" ➔ "Settings" ➔ "Compiler" ➔ "Input". In "-i (Include User Search Paths (-i)" and after "lwgpio" or at the bottom of the list is "lwgpio" is not found add:
   `"${MQX_ROOT_DIR}/mqx/source/io/lwadc"`
   - **NOTE:** This also applies for "Release" versions of the projects

2. Open "Properties" for project and go to "C/C++ Build" ➔ "Settings" ➔ "Librarian" and set the "Model" to c9x

Application Notes:

1. Change the USB Device Descriptor ("devDesc") statement in **j20_Test_##/Sources/config/config_usbd_config.c** (where "##" is the program release number to:

```
/* Device descriptor */
static const unsigned char devDesc[] = {
    18,      /* bLength */
    1, /* bDescriptorType */
    0x10, 0x01,     /* bcdUSB */
    0x0,     /* bDeviceClass */
    0x0,     /* bDeviceSubClass */
    0x0,     /* bDeviceProtocol */
    0x8,     /* bMaxPacketSize */
    0x07, 0x29,     /* idVendor */  /*  Mimetics USB Vendor ID  */
    0x01, 0x00,     /* idProduct */  /*  Mimetics Product ID  */
   0x00, 0x00,      /* bcdDevice */
    2, /* iManufacturer */
    1, /* iProduct */
    3, /* iSerialNumber */
    1, /* bNumConfigurations */
    0   /* no M$ OS string */
};     /* eof devDesc */
```

Mimetics Robot user_config.h

```
/**HEADER*********************************************************************
*
* Copyright (c) 2008 Freescale Semiconductor;
* All Rights Reserved
*
*****************************************************************************
*
* THIS SOFTWARE IS PROVIDED BY FREESCALE "AS IS" AND ANY EXPRESSED OR
* IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
* OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
* IN NO EVENT SHALL FREESCALE OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT,
* INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
* (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
* SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
* HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
* STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
* IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF
* THE POSSIBILITY OF SUCH DAMAGE.
*
*****************************************************************************
*
* $FileName: user_config.h$
* $Version : 3.8.4.0$
* $Date    : Sep-18-2012$
*
* Comments:
*
*   User configuration for MQX components
*
*END*********************************************************************/

#ifndef __user_config_h__
#define __user_config_h__

/* mandatory CPU identification */
#define MQX_CPU                 PSP_CPU_MK20DN512

/* MGCT: <generated code> */
#define BSPCFG_ENABLE_I2C0                      0
#define BSPCFG_ENABLE_I2C1                      0
#define BSPCFG_ENABLE_II2C0                     0
#define BSPCFG_ENABLE_II2C1                     0
#define BSPCFG_ENABLE_II2C0_FB                  1
#define BSPCFG_ENABLE_II2C1_FB                  1
#define BSPCFG_ENABLE_RTCDEV                    1
#define BSPCFG_ENABLE_PCFLASH                   0
#define BSPCFG_ENABLE_SPI0                      0
#define BSPCFG_ENABLE_ISPI0                     0
#define BSPCFG_ENABLE_SPI1                      0
```

```
#define BSPCFG_ENABLE_ISPI1                     0
#define BSPCFG_ENABLE_SPI2                      0
#define BSPCFG_ENABLE_ISPI2                     0
#define BSPCFG_ENABLE_ADC0                      0
#define BSPCFG_ENABLE_ADC1                      0
#define BSPCFG_ENABLE_LWADC0                    1
#define BSPCFG_ENABLE_LWADC1                    1
#define BSPCFG_ENABLE_FLASHX                    1
#define BSPCFG_ENABLE_CRC                       0
#define BSPCFG_ENABLE_IODEBUG                   0
#define BSPCFG_ENABLE_SAI                       1

#define BSPCFG_ENABLE_TTYA                      0
#define BSPCFG_ENABLE_ITTYA                     1

#define BSPCFG_ENABLE_TTYB                      0
#define BSPCFG_ENABLE_ITTYB                     1

#define BSPCFG_ENABLE_TTYC                      0
#define BSPCFG_ENABLE_ITTYC                     0

#define BSPCFG_ENABLE_TTYD                      0
#define BSPCFG_ENABLE_ITTYD                     0

#define BSPCFG_ENABLE_TTYE                      0
#define BSPCFG_ENABLE_ITTYE                     0

#define BSPCFG_ENABLE_TTYF                      0
#define BSPCFG_ENABLE_ITTYF                     0

#define BSP_DEFAULT_INTERRUPT_STACK_SIZE        (1024L)
#define BSP_DEFAULT_MAX_MSGQS                   (44L)
#define BSP_DEFAULT_IO_CHANNEL                  NULL

#define MQX_USE_IDLE_TASK                       1

#define MQX_USE_TIMER                           1

#define MQX_TASK_DESTRUCTION                    1
#define MQX_COMPONENT_DESTRUCTION               1


//  todo: Changed 2013.11.08 - MAP
#define MQX_ENABLE_LOW_POWER                    1


#define RTCSCFG_ENABLE_ICMP                     1
#define RTCSCFG_ENABLE_SNMP                     0
#define RTCSCFG_ENABLE_UDP                      1
#define RTCSCFG_ENABLE_TCP                      1
#define RTCSCFG_ENABLE_STATS                    1
#define RTCSCFG_ENABLE_GATEWAYS                 1
```

```
#define FTPDCFG_USES_MFS                        1

#define MQX_USE_TIMER                           1

#define TELNETDCFG_NOWAIT                       FALSE

#define HTTPDCFG_POLL_MODE                      0
#define HTTPDCFG_STATIC_TASKS                   0
#define HTTPDCFG_DYNAMIC_TASKS                  1
/* MGCT: </generated_code> */

/*
** include common settings
*/

/* use the rest of defaults from small-RAM-device profile */
#include "small_ram_config.h"

/* and enable verification checks in kernel */
#include "verif_enabled_config.h"

#endif /* __user_config_h__ */
```

z1st_LED_DEMO Test Application:

z1st_LED_DEMO.c:

```
/**************************************************************************
*
*   This file contains MQX only stationery code.
*
**************************************************************************/
#include "z1st_LED_DEMO.h"


#if !BSPCFG_ENABLE_IO_SUBSYSTEM
#error This application requires BSPCFG_ENABLE_IO_SUBSYSTEM defined non-zero
in user_config.h. Please recompile BSP with this option.
#endif


#ifndef BSP_DEFAULT_IO_CHANNEL_DEFINED
#error This application requires BSP_DEFAULT_IO_CHANNEL to be not NULL.
Please set corresponding BSPCFG_ENABLE_TTYx to non-zero in user_config.h and
recompile BSP with this option.
#endif



//  LED Global Variables
LWGPIO_STRUCT               blueLED1;
LWGPIO_STRUCT               blueLED2;

LWGPIO_STRUCT               greenLED1;
LWGPIO_STRUCT               greenLED2;
LWGPIO_STRUCT               greenLED3;
LWGPIO_STRUCT               greenLED4;
LWGPIO_STRUCT               greenLED5;

LWGPIO_STRUCT               greenLED6;


TASK_TEMPLATE_STRUCT MQX_template_list[] =
{
/*  Task number, Entry point, Stack, Pri, String, Auto? */
   {MAIN_TASK,   Main_task,   1500,  9,   "main", MQX_AUTO_START_TASK},
   {0,           0,           0,     0,   0,      0,                 }
};


/*TASK*-----------------------------------------------------
*
* Task Name    : Main_task
* Comments     :
*    This task prints "1st_LED_DEMO" (Project Name &
*    has an LED light pattern
```

```
*
*    For the Mimetics Jade Robot
*
*END*----------------------------------------------------------*/

void Main_task(uint_32 initial_data) {
int                         i;
FILE_PTR                    fh_ptr;


      if(NULL == (fh_ptr = fopen("iodebug:", NULL))) {
            printf("Cannot open the debug output\n");
            fflush(stdout);
      } else {
            _io_set_handle(IO_STDOUT, fh_ptr);
      }

      printf("\n1st_LED_DEMO\n");
      fflush(stdout);


      if (!lwgpio_init(&blueLED1, LWGPIO_PORT_C | LWGPIO_PIN16,
                              LWGPIO_DIR_OUTPUT,
                              LWGPIO_VALUE_HIGH)) {
            printf("Initializing blueLED1 as output failed.\n");
            fflush(stdout);
            _task_block();
      }
      lwgpio_set_functionality(&blueLED1, LWGPIO_MUX_C16_GPIO);
      if (!lwgpio_init(&blueLED2, LWGPIO_PORT_C | LWGPIO_PIN17,
                              LWGPIO_DIR_OUTPUT,
                              LWGPIO_VALUE_HIGH)) {
            printf("Initializing blueLED2 as output failed.\n");
            fflush(stdout);
            _task_block();
      }
      lwgpio_set_functionality(&blueLED2, LWGPIO_MUX_C17_GPIO);


      if (!lwgpio_init(&greenLED1, LWGPIO_PORT_C | LWGPIO_PIN11,
                              LWGPIO_DIR_OUTPUT,
                              LWGPIO_VALUE_HIGH)) {
            printf("Initializing greenLED1 as output failed.\n");
            fflush(stdout);
            _task_block();
      }
      lwgpio_set_functionality(&greenLED1, LWGPIO_MUX_C11_GPIO);
      if (!lwgpio_init(&greenLED2, LWGPIO_PORT_C | LWGPIO_PIN13,
                              LWGPIO_DIR_OUTPUT,
                              LWGPIO_VALUE_HIGH)) {
            printf("Initializing greenLED2 as output failed.\n");
            fflush(stdout);
            _task_block();
      }
```

```
lwgpio_set_functionality(&greenLED2, LWGPIO_MUX_C13_GPIO);
if (!lwgpio_init(&greenLED3, LWGPIO_PORT_C | LWGPIO_PIN15,
                        LWGPIO_DIR_OUTPUT,
                        LWGPIO_VALUE_HIGH)) {
      printf("Initializing greenLED3 as output failed.\n");
      fflush(stdout);
      _task_block();
}
lwgpio_set_functionality(&greenLED3, LWGPIO_MUX_C15_GPIO);
if (!lwgpio_init(&greenLED4, LWGPIO_PORT_C | LWGPIO_PIN14,
                        LWGPIO_DIR_OUTPUT,
                        LWGPIO_VALUE_HIGH)) {
      printf("Initializing greenLED4 as output failed.\n");
      fflush(stdout);
      _task_block();
}
lwgpio_set_functionality(&greenLED4, LWGPIO_MUX_C14_GPIO);
if (!lwgpio_init(&greenLED5, LWGPIO_PORT_C | LWGPIO_PIN12,
                        LWGPIO_DIR_OUTPUT,
                        LWGPIO_VALUE_HIGH)) {
      printf("Initializing greenLED5 as output failed.\n");
      fflush(stdout);
      _task_block();
}
lwgpio_set_functionality(&greenLED5, LWGPIO_MUX_C12_GPIO);
if (!lwgpio_init(&greenLED6, LWGPIO_PORT_E | LWGPIO_PIN3,
                        LWGPIO_DIR_OUTPUT,
                        LWGPIO_VALUE_HIGH)) {
      printf("Initializing greenLED6 as output failed.\n");
      fflush(stdout);
      _task_block();
}
lwgpio_set_functionality(&greenLED6, LWGPIO_MUX_E3_GPIO);


_time_delay(1000);


lwgpio_set_value(&blueLED1, LWGPIO_VALUE_LOW);
lwgpio_set_value(&blueLED2, LWGPIO_VALUE_LOW);

lwgpio_set_value(&greenLED1, LWGPIO_VALUE_LOW);
lwgpio_set_value(&greenLED2, LWGPIO_VALUE_LOW);
lwgpio_set_value(&greenLED3, LWGPIO_VALUE_LOW);
lwgpio_set_value(&greenLED4, LWGPIO_VALUE_LOW);
lwgpio_set_value(&greenLED5, LWGPIO_VALUE_LOW);
lwgpio_set_value(&greenLED6, LWGPIO_VALUE_LOW);


for (i = 0 ; 8 > i; ++i) {

      _time_delay(500);
      lwgpio_set_value(&greenLED6, LWGPIO_VALUE_LOW);
      lwgpio_set_value(&blueLED1, LWGPIO_VALUE_HIGH);
```

```
            _time_delay(333);
            lwgpio_set_value(&blueLED1, LWGPIO_VALUE_LOW);
            lwgpio_set_value(&blueLED2, LWGPIO_VALUE_HIGH);

            _time_delay(333);
            lwgpio_set_value(&blueLED2, LWGPIO_VALUE_LOW);
            lwgpio_set_value(&greenLED1, LWGPIO_VALUE_HIGH);

            _time_delay(500);
            lwgpio_set_value(&greenLED1, LWGPIO_VALUE_LOW);
            lwgpio_set_value(&greenLED2, LWGPIO_VALUE_HIGH);

            _time_delay(500);
            lwgpio_set_value(&greenLED2, LWGPIO_VALUE_LOW);
            lwgpio_set_value(&greenLED3, LWGPIO_VALUE_HIGH);

            _time_delay(500);
            lwgpio_set_value(&greenLED3, LWGPIO_VALUE_LOW);
            lwgpio_set_value(&greenLED4, LWGPIO_VALUE_HIGH);

            _time_delay(500);
            lwgpio_set_value(&greenLED4, LWGPIO_VALUE_LOW);
            lwgpio_set_value(&greenLED5, LWGPIO_VALUE_HIGH);

            _time_delay(500);
            lwgpio_set_value(&greenLED5, LWGPIO_VALUE_LOW);
            lwgpio_set_value(&greenLED6, LWGPIO_VALUE_HIGH);
        }

        _time_delay(500);
        lwgpio_set_value(&greenLED6, LWGPIO_VALUE_LOW);

        _task_block();

}

/* EOF */
```

z1st_LED_DEMO.h:

```
#ifndef __Z1st_LED_DEMO_h_
#define __Z1st_LED_DEMO_h_
#include <mqx.h>
#include <bsp.h>

#define MAIN_TASK 1

extern void Main_task(uint_32);

/* PPP device must be set manually and
** must be different from the default IO channel (BSP_DEFAULT_IO_CHANNEL)
*/
#define PPP_DEVICE      "ittyb:"

/*
** Define PPP_DEVICE_DUN only when using PPP to communicate
** to Win9x Dial-Up Networking over a null-modem
** This is ignored if PPP_DEVICE is not #define'd
*/
#define PPP_DEVICE_DUN  1

/*
** Define the local and remote IP addresses for the PPP link
** These are ignored if PPP_DEVICE is not #define'd
*/
#define PPP_LOCADDR     IPADDR(192,168,0,216)
#define PPP_PEERADDR    IPADDR(192,168,0,217)

/*
** Define a default gateway
*/
#define GATE_ADDR       IPADDR(192,168,0,1)

#endif /* __Z1st_LED_DEMO_h_ */
```

Document Updates:

2013.07.28 – Initial Release with basic MK20DN512VLL10 BSP
2013.07.29 – Rebuild with MK60DN512VLL10 with twrk60n512 (using the twrk40d100m project for clocking information) rather than just twrk20d72m
– Changed project from "r20" to "j20"
– Updated to include updated i2c Driver
– Reviewed Device Memory Initialization and put in corrections
2013.07.31 – Evaluating the operation of the I2C driver
– Added "BSP Modifications for Mimetics Robot" as a section. **NOTE:** this section should be ignored unless using the Mimetics robot or trying to understand how to modify a BSP for a specific product
2013.08.03 – Added lwadc to the BSP
– Added i2c_int_k_fb.c as an object to this Word file
– Added Mimetics robot user_config.h
2013.09.07 – Noted that UART1 in the product is the same as the BSP default
2013.09.25 – Added BSP modification for 2ms clock tick instead of the standard 5ms tick
2013.11.17 – Added BSP modification for Wake Up on "Down Button"
– Added MQX_ENABLE_LOW_POWER in user_config.h
2013.11.18 – Updated "init_sci.c" to not shut down "ittya:" and "ittyb:" due to "unhandled interrupt" error during execution.
2014.01.05 – Updated "Flashx" code to support "swap" in CW 4.0
2014.02.14 – Updated "I^2C" code to support I2C0
– Added MQX_USE_TIMER in user_config.h
– Updated BSP_DEFAULT_MAX_MSGQs in user_config.h
2014.02.27 – Updated to include high speed camera updates
2014.04.02 – Updated to include changes to the high speed camera interface
2014.04.12 – Added "Application Notes"
– Indicated where the Mimetics VID and PID is to be placed (USB Device Descriptor)
2014.04.20 – Added "USB Update" Section
2018.09.07 – Added Martin Látal's Instructions for setting up a custom bsp
2018.09.08 – Updated Logo at top of page
– Updated Start of document to reflect what was happening in 2018 as I tried to recreate the bsp
– Cleaned up some formatting in the document
– Added commenting out forced assembly language in dispatch.s of the psps
– Added Updates according to what was discovered in setting up the bsp
2018.09.09 – Streamlining the operations
– Noted that this is for CodeWarrior 11.0
2018.09.10 – Continuing to streamline the operations
– Fixed up how to clean up .project and .cproject
– Noted differences between processors in the twrk60d100m and the Jade Robot
– Added "1st_LED_DEMO" as the first test application
2018.09.11 – Continuing to streamline the operations from I^2C Driver Install
– Taking out ALL SD Card Options

2018.09.11 – Continuing to streamline the operations from lwadc Driver Install
2018.09.12 – Continuing to streamline the operations from lwadc Driver Install
2018.09.14 – Continuing to streamline the operations from lwadc Driver Install
2018.09.16 – Continuing to streamline the operations from lwadc Driver Install
– Changed Test projects to start with "z" so they are built AFTER RTOS Projects
2018.09.16 – Continuing to streamline the operations and look at I2C operation
– Added note that twrk40d100m and twrk21d50m bsps are used for clocking and lwadc
– Removed Martin Látal's Instructions for setting up a custom bsp/Don't think it's needed
2018.09.16 – Continuing to streamline the operations and start on Low Power Mode
– Work through "Flashx" changes
– Work through "BSP Modifications for Jade Robot"
2018.09.20 – Continuing to streamline the operations and work on BSP Modifications for High Speed Camera Operation
– Trying to build a j20_Test_## Project and adding changes list to document
– Replaced modifying "serl_pol_kuart.c" & "serl_int_kuart.c" with providing a file to copy into the bsp
2018.09.26 – Clean up document so it just relates to the BSP. Got rid of "USB Updates" as it wasn't relevant to the bsp build
– Removed "Final Notes" which were out of date and resolved
– Removed Notes about 2nd_OLED_DEMO project as this is no longer relevant