

```

/*****
MPLAB Harmony Application Source File

Company:
    Microchip Technology Inc.

File Name:
    app.c

Summary:
    This file contains the source code for the MPLAB Harmony application.

Description:
    This file contains the source code for the MPLAB Harmony application.
    implements the logic of the application's state machine and it may call
    API routines of other MPLAB Harmony modules in the system, such as driv
    system services, and middleware. However, it does not call any of the
    system interfaces (such as the "Initialize" and "Tasks" functions) of a
    the modules in the system or make any assumptions about when those func
    are called. That is the responsibility of the configuration-specific s
    files.
*****/

// DOM-IGNORE-BEGIN
/*****
Copyright (c) 2013-2014 released Microchip Technology Inc. All rights rese

Microchip licenses to you the right to use, modify, copy and distribute
Software only when embedded on a Microchip microcontroller or digital signa
controller that is integrated into your product or third party product
(pursuant to the sublicense terms in the accompanying license agreement).

You should refer to the license agreement accompanying this Software for
additional information regarding your rights and obligations.

SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIN
EITHER EXPRESS OR IMPLIED, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF
MERCHANTABILITY, TITLE, NON-INFRINGEMENT AND FITNESS FOR A PARTICULAR PURPO
IN NO EVENT SHALL MICROCHIP OR ITS LICENSORS BE LIABLE OR OBLIGATED UNDER
CONTRACT, NEGLIGENCE, STRICT LIABILITY, CONTRIBUTION, BREACH OF WARRANTY, O
OTHER LEGAL EQUITABLE THEORY ANY DIRECT OR INDIRECT DAMAGES OR EXPENSES
INCLUDING BUT NOT LIMITED TO ANY INCIDENTAL, SPECIAL, INDIRECT, PUNITIVE OR
CONSEQUENTIAL DAMAGES, LOST PROFITS OR LOST DATA, COST OF PROCUREMENT OF
SUBSTITUTE GOODS, TECHNOLOGY, SERVICES, OR ANY CLAIMS BY THIRD PARTIES

```

\*\*\*\*\*

It

ers,

ny of  
tions  
ystem

\*\*\*\*\*/

\*\*\*\*\*  
rved.

l

D,

SE.

R

```
(INCLUDING BUT NOT LIMITED TO ANY DEFENSE THEREOF), OR OTHER SIMILAR COSTS.
*****
// DOM-IGNORE-END

// *****
// *****
// Section: Included Files
// *****
// *****

#include "app.h"
#include "Mc32DriverLcd.h"

// *****
// *****
// Section: Global Data Definitions
// *****
// *****

// *****
/* Application Data

Summary:
    Holds application data

Description:
    This structure holds the application's data.

Remarks:
    This structure should be initialized by the APP_Initialize function.

    Application strings and buffers are be defined outside this structure.
*/

APP_DATA appData;
S_ADCCConvert ValTemp;
// *****
// *****
// Section: Application Callback Functions
// *****
// *****

/* TODO: Add any necessary callback functions.
```

\*\*\*\*\* /

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

```

*/

// *****
// *****
// Section: Application Local Functions
// *****
// *****

/* TODO: Add any necessary local functions.
*/

// *****
// *****
// Section: Application Initialization and State Machine Functions
// *****
// *****

/*****
Function:
    void APP_Initialize ( void )

Remarks:
    See prototype in app.h.
*/

void APP_Initialize ( void )
{
    /* Place the App state machine in its initial state. */
    appData.state = APP_STATE_INIT;

    /* TODO: Initialize your application's state machine and other
     * parameters.
     */
}

/*****
Function:
    void APP_Tasks ( void )

Remarks:

```

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

\* \* \* \* \*

```

    See prototype in app.h.
*/

void APP_Tasks ( void )
{
    /* Check the application's current state. */
    switch ( appData.state )
    {
        /* Application's initial state. */
        case APP_STATE_INIT:
        {
            // Initialisation de l'LCD
            lcd_init();
            lcd_gotoxy(1,1);
            printf_lcd("PROJ REGTHERM");
            lcd_gotoxy(1,2);
            printf_lcd("Neziri Taulant");
            lcd_bl_on();

            // Démarrage des Timers
            DRV_TMR0_Start();
            DRV_TMR1_Start();
            DRV_TMR2_Start();

            // Démarrage des OC
            DRV_OC0_Start();
            DRV_OC1_Start();

            // Life LED On
            LED_VOn();

            //ADC Init
            BSP_InitADC10();

            appData.state = APP_STATE_WAIT;

            break;
        }

        case APP_STATE_SERVICE_TASKS:
        {
            ValTemp = ValConvert();
            float Temp = ValTemp.Val_T;

```





```
        //float Temp = ValTemp.Val_T;

        lcd_gotoxy(1,4);
        printf_lcd("ValRaw : %2.2f [C]", Temp);

        appData.state = APP_STATE_WAIT;
        break;
    }

    /* TODO: implement your application state machine.*/
    case APP_STATE_WAIT:
    {
        // Ne rien faire ici
        break;
    }

    /* The default state should never be executed. */
    default:
    {
        /* TODO: Handle error in application's state machine. */
        break;
    }
}

// Fonction de mise à jour du switch
void APP_UpdateState (APP_STATES NewState)
{
    appData.state = NewState;
}

/*****
End of File
*/
```

\*\*\*\*\*