

2013

Underwater Equipment for Physiological Monitoring Development Board

USER MANUAL



Copyright ©
Bogazici Uluslararası Eğitim Dan Mrk
ve Ltd Ş
04.09.2013

WARNING!

The UEPM Development Board is a development tool. It should not be used as a qualified diving equipment but as a tool to develop and test diving equipment.

WARNING!

Do not use the UEPM Development Board hardware and software, before reading user manual. Otherwise, you could cause accidents resulting in death or serious injuries.

WARNING!

The UEPM Development Board is equipped with Lithium batteries that may cause fire and / or explosion in case of any inappropriate usage. Read the user manual carefully to understand how to use the UEPM Development Board.

UEPM DEVELOPMENT BOARD

1. Precautions	5
2. Introduction.....	6
3. Board Overview	7
4. Functions.....	9
4.1. Target Board.....	9
4.2. LCD Display.....	10
4.3. Extension Pins	11
4.4. Battery and Charging Circuit	11
4.5. Control Pins	13
4.6. Switches	14
4.7. Serial Communication.....	15
4.8. INNOVASUB Pressure Sensor	18
4.9. Piezo-Switches	19
4.10. Potentiometers.....	20
4.11. LEDs.....	20
4.12. Oxygen Sensor	21
4.13. General Purpose ADC Inputs.....	22
4.14. MS5541C Pressure Sensor	23
5. Programs.....	24
5.1. Sample Codes	24
5.2. USB Sample Code.....	24
5.3. SD Card Sample.....	25
5.4. UEPM Sample Code.....	25
Configuration.....	26
Mapping	26
Main	26
Timer.....	27
LEDs	27
PowerManager.....	27
LCD.....	28
UART	28
XBee	28
Piezo	28
SD Slot.....	28
ADC	29
Pressure Sensor	29
Interrupts.....	29

Table of Figures

Figure 1 – Board Overview.....	7
Figure 2 – MSP-TS430PN80USB Target Board.....	9
Figure 3 – LCD Display	10
Figure 4 – Extension Pins	11
Figure 5 – Battery	12
Figure 6 – Mini-USB Connector for Charging the Battery	12
Figure 7 – Control Pins (on the Left)	13
Figure 8 – SW1 and SW2	14
Figure 9 – SW3 and L2	15
Figure 10 – Jumper JP9 in Position SNS1 and Jumper JP10 in Position SNS2	16
Figure 11 – XBEE1 (RN-171XV) and XBEE2 (RN-42XV).....	16
Figure 12 – SENSOR2, SENSOR1 and Micro-SD Slot	17
Figure 13 – Underwater Wireless Module Connector	17
Figure 14 – Two INNOVASUB Pressure Sensors and Two T-Pipes.....	18
Figure 15 –INNOVASUB Pressure Sensors with USB Cable for Connection with UEPM Development Board	19
Figure 16 – PIEZO1 and PIEZO2 Piezo-Switches.....	19
Figure 17 – POT1 and POT2 Potentiometers.....	20
Figure 18 – LED1, LED2, LED3 and LED4.....	21
Figure 19 – X4 (Jack 3.5mm) and JP15 (0.1”) Connectors for Oxygen Sensor	21
Figure 20 – ADC Module – JP16 Connector for 6 x 12-bit ADC Inputs.....	22
Figure 21 – SENSOR3.....	23

UEPM Development Board

1. PRECAUTIONS

1. **CAUTION!** Keep the UEPM Development Board away from water or wet surfaces.
2. **CAUTION!** Lay the UEPM Development Board on a flat surface. Any fall or shock could permanently damage the unit and thus void warranty.
3. **CAUTION!** Keep the UEPM Development Board away from direct sunlight. Even in cold environments, the sunlight rays can be powerful enough to damage the LCD display or others components, or cause the overheating of some parts of the circuits.
4. **CAUTION!** Use only the power supply provided with the UEPM Development Board or an appropriate 5V and >1000mA power supply. An inappropriate power supply could permanently damage the UEPM Development Board.
5. **DANGER!** Follow the temperature usage recommended. The UEPM Development Board is equipped with a lithium battery, used for its high capacity. However, lithium battery type can be dangerous if not properly used. The battery may explode if not charged properly (wrong power supply) or over heated (hot weather). As well, the battery may lose its capacity if used in cold conditions.
6. **CAUTION!** USB connections 5 and 10 shown in Figure 1 **Hata! Başvuru kaynağı bulunamadı.** are not standard USB compliant. You should only connect sensors delivered with the UEPM Development Board to these connections. Otherwise you could damage the device and this would void the warranty.
7. **CAUTION!** Sensors should be used only with air and dry environment. Using sensor for underwater pressure or using sensors with different gases could cause permanent damage on sensors.
8. **DANGER!** Do not use the UEPM Development Board hardware and software before reading user manual. Otherwise, you could cause accidents resulting in death or injuries.

2. INTRODUCTION

The Underwater Equipment for Physiological Monitoring Development Board aims at helping research and development in diving technologies. It combines all the hardware and software required for monitoring the diver's physiology underwater.

The UEPM Development Board is provided with all the accessories, documents and sample code to start developing immediately.

This document describes the hardware components of the UEPM Development Board as well as their associated sample code.

3. BOARD OVERVIEW

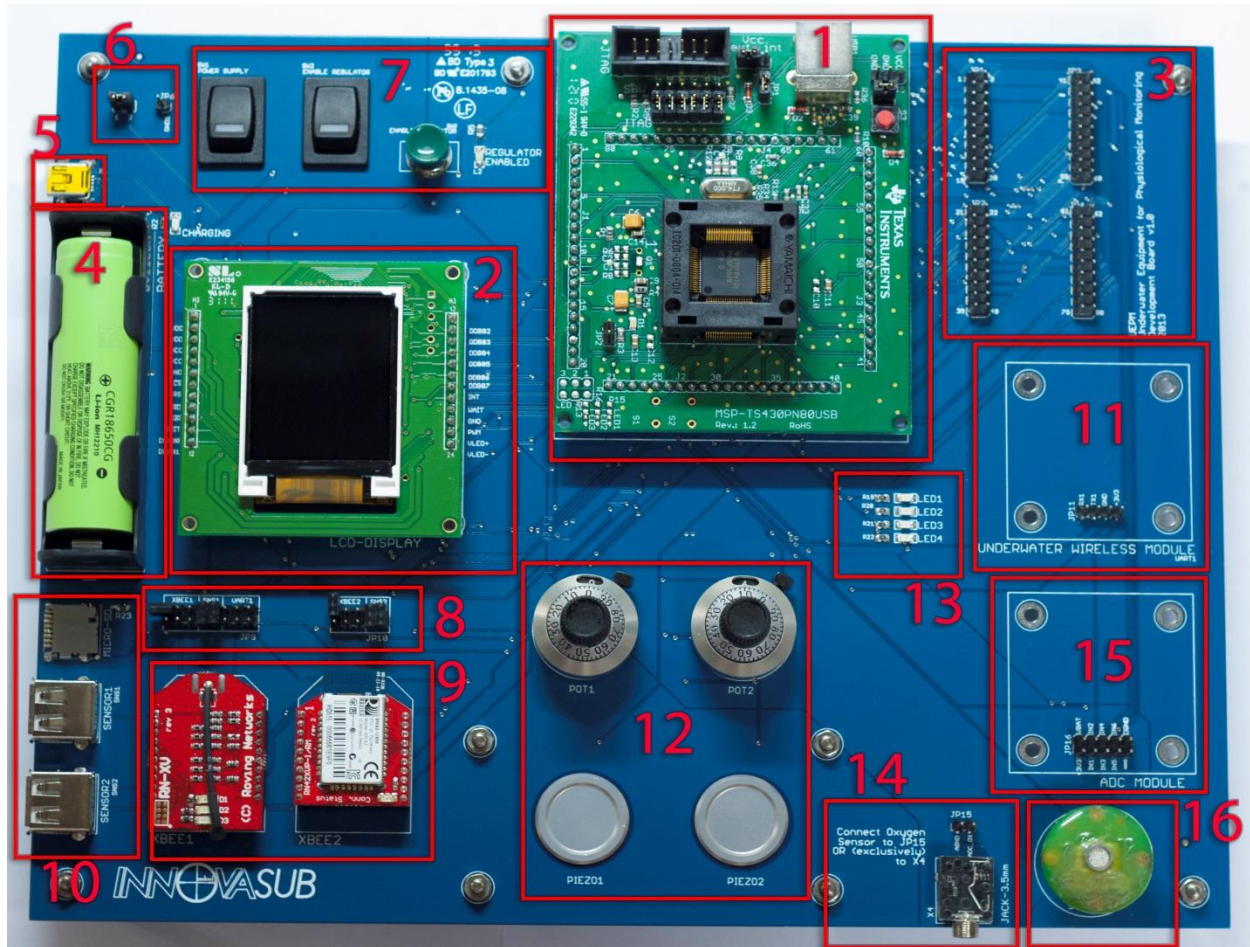


Figure 1 – Board Overview

- 1 MSP-TS430PN80USB. Target Board for Texas Instruments 80-PN package MSP430 microcontrollers. Equipped with JTAG for on-board programming and debugging, and USB. Specific datasheet available on the CD provided with the UEPM Development Board.
- 2 INT018ATFT LCD display. 1.8" TFT, 160x128 RGB integrated driver. Specific datasheet available on the CD provided with the UEPM Development Board.
- 3 Extension of the MSP430's 80 pins. 0.1" pitch connectors.
- 4 CGR18650CG. Lithium-Ion Rechargeable battery.
- 5 Mini-USB connector. Power supply to recharge the battery. NOT USB COMPLIANT. Connect only a power supply with 5V and at least 1A capacity.
- 6 Reference pin (GND) and pins to control current through the battery.
- 7 Switches controlling power supply to the MSP430 and peripheral components.
- 8 Jumpers to select the devices to be connected to the two UART ports of the MSP430. For each port (UART1 and UART2) can be connected an XBEE port, a pressure sensor or (only on UART1) the Underwater Wireless Module.
- 9 XBEE1 and XBEE2 ports. 2 x Ports compatible with XBee footprint modules.

- 10 2 x USB Female A connectors for connection with Innovasub Pressure sensor. Connectors NOT USB COMPLIANT. Do not connect any USB device to these connectors.
1 x Micro-SD socket. Connected to the SPI port of the MSP430.
- 11 Underwater Wireless Module, connected to the UART1 port (USCI0) of the MSP430.
- 12 2 x High precision, 10-turn potentiometer. Connected to 12-bit ADC of the MSP430.
2 x Piezo-electric switches. Connected to digital, interruptible input of the MSP430.
- 13 4 x General purpose LEDs. Connected to digital output of the MSP430.
- 14 Port for connection with Oxygen Sensor. Jack 3.5mm or standard 0.1" connectors available. DO NOT USE BOTH CONNECTORS AT THE SAME TIME as they are connected to the same 12-bit ADC input.
- 15 General purpose 12-bit ADC acquisition. Connection for custom board.
6 x ADC inputs, power from battery and regulated 3.3V, Digital and Analog grounds.
- 16 MS5541C pressure sensor mounted on INNOVASUB PCB.

4. FUNCTIONS

This paragraph details every hardware function of the UEPM Development Board..

4.1. Target Board

The Texas Instrument's MSP-TS430PN80USB target board is compatible with all MSP430 in 80-PN package. However, the UEPM Development Board was designed and optimized for the specific MSP430F5529. It is delivered with this microcontroller already pre-loaded with the UEPM Sample Program.

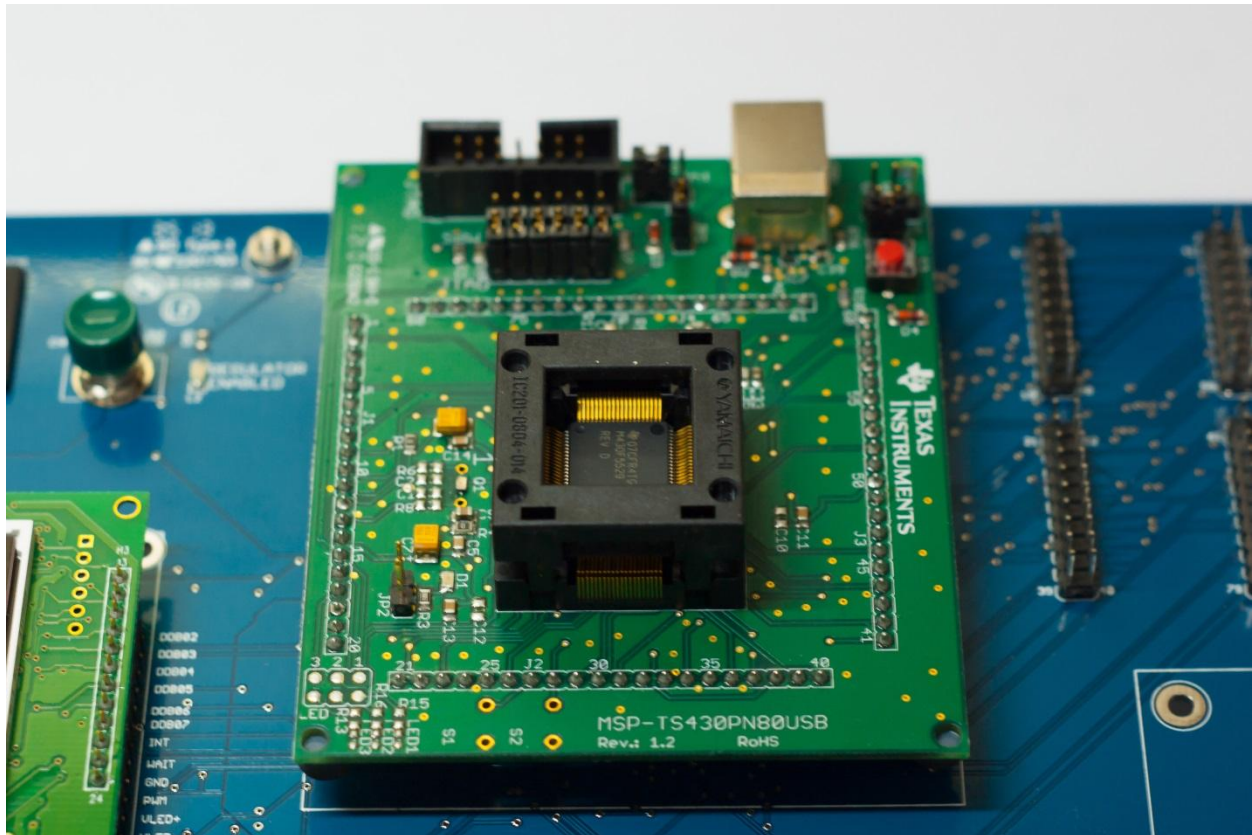


Figure 2 – MSP-TS430PN80USB Target Board

It is recommended to use the UEPM Development Board with MSP430F5529 as some functions might not be available with other references of microcontroller.

Leave jumper JP2 open as the pin P1.0 is already used on the UEPM Development Board.

The target board has a 32,768 Hz crystal soldered on XT1 for keeping a correct track of time and a 4 MHz crystal soldered on XT2 used for USB. The target board is also

equipped with USB connector. Please refer to *5.2 USB Sample Code* for details on USB program.

In order to load a program to the MSP430 or to run a debug, place the target board's JP3 in 'ext' position and close the UEPM Development Board's switches SW1 and SW2. Make sure the battery has enough charge to power the board as it is powering the whole board. Connect the MSP-FET430UIF provided with the UEPM Development Board to the JTAG connector of the target board and connect its USB cable to your computer. From Code Composer Studio, you can now run a program or debug.

Note: the UEPM Development Board is equipped with several peripherals that may consume important current, especially if a Wi-Fi module is plugged to the XBEE connector and the LCD screen is on. This is why the power must be supplied by the battery instead of the MSP-FET430UIF while debugging.

4.2. LCD Display

The INT018ATFT is 1.8" TFT display of 160 x 128. It uses 8 data pins for optimum display time. Please refer the device's datasheet for further information.



Figure 3 – LCD Display

4.3. Extension Pins

The 80 extension pins are the extension of the MSP430's 80 pins. Numbered from 1 to 80 in the same order as the MSP430. They use four 2x10 standard 0.1" pitch connectors allowing further test and development.

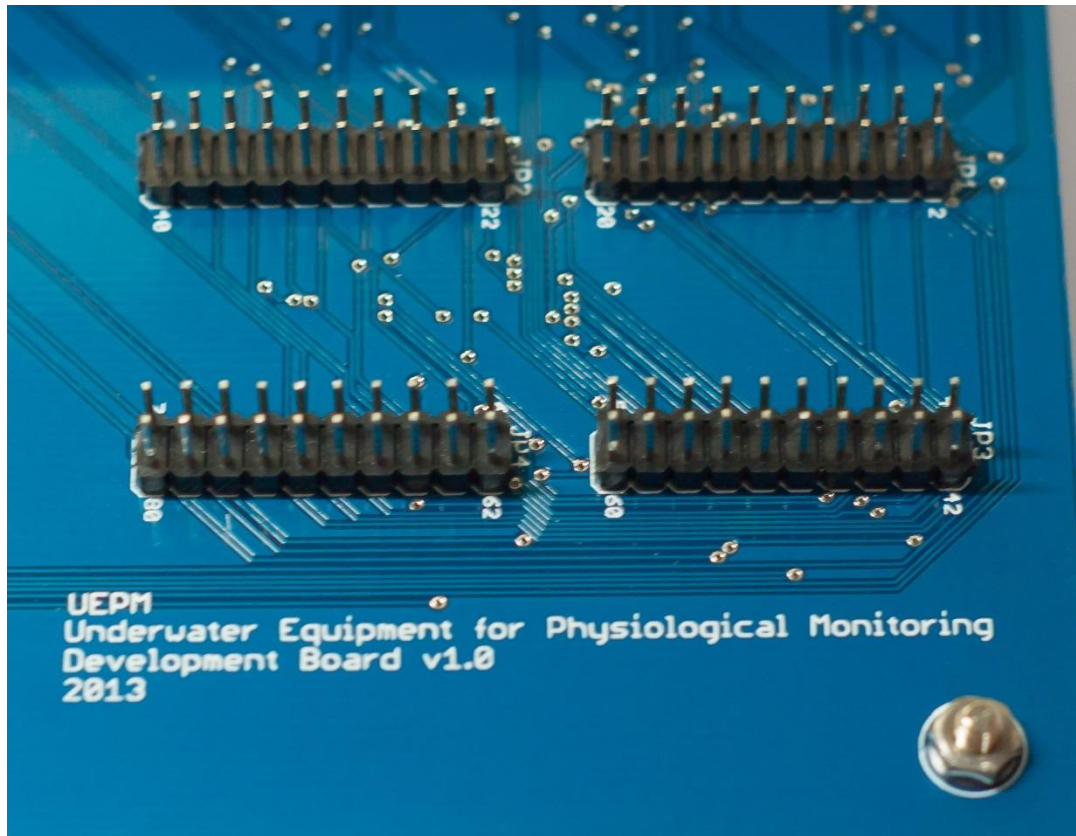


Figure 4 – Extension Pins

4.4. Battery and Charging Circuit

The battery used on the board is CGR18650CG. It is a Rechargeable, single-cell Lithium-Ion battery, with a capacity of 2000 mAh. Its nominal voltage is 3.7V but it can vary between 4.2V and 2.7V depending on its state-of-charge. Its package is the standard 18650.

The UEPM Development Board is equipped with a charging circuit to manage the charge of the battery using a Constant Current – Constant Voltage method. The maximum current provided to the battery is 500mA and the maximum voltage is 4.2V. When charging, the LED CHARGING is bright red.

In order to properly proceed to the charge of the battery, use a power supply of 5V and a minimum capacity of 1000mA. It is highly recommended that you use the power supply provided with the UEPM Development Board.

The connector used for the power supply is a Mini-USB female connector. IT IS NOT USB COMPLIANT. Therefore, do not connect any USB device to that port. You may only plug a power supply.



Figure 5 – Battery

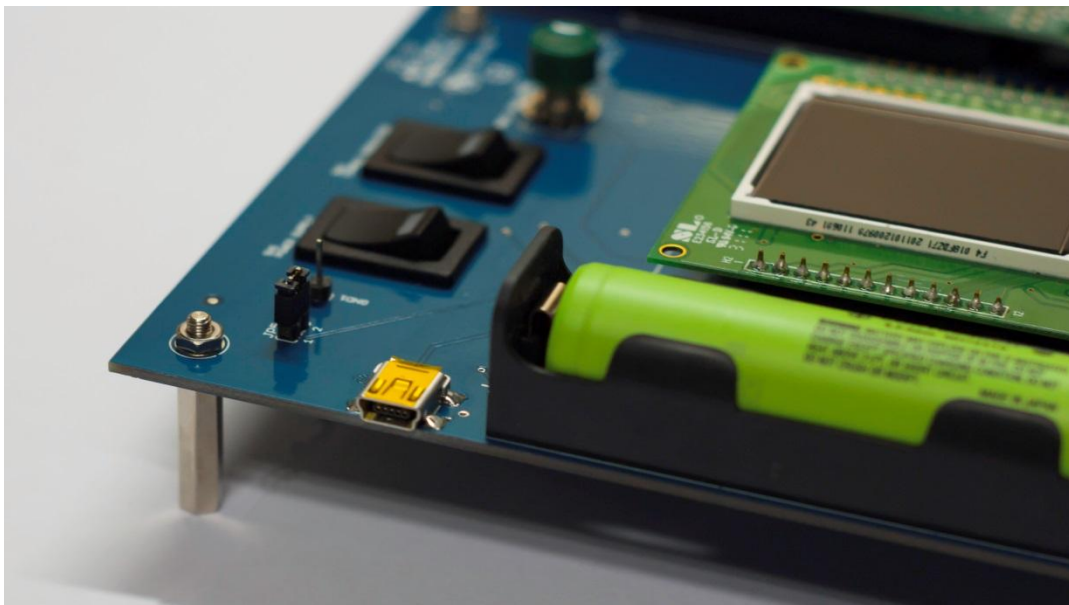


Figure 6 – Mini-USB Connector for Charging the Battery

CAUTION!

Lithium batteries are explosive and can easily fire. The battery may explode or fire if mistreated. Do not disassemble or dispose on fire. Do not charge out of charging conditions. Do not heat above 100°C (217°F) or short circuit. Do not crush or modify.

CAUTION!

The Mini-USB connector used for charging the battery is not USB compliant. Do not connect anything else than the power supply to that connector. Connecting a USB device to that port may damage it, cause explosion or fire.

CAUTION!

Use a power supply of regulated 5V and at least 1000mA capacity. The use of an inappropriate power supply may cause explosion, fire or the damage of the power supply and / or the battery.

CAUTION!

Do not charge the battery out of the 5°C – 45°C (41°F – 113°F) temperature range.

4.5. Control Pins

The control pins can be used to control the voltage on the board and to control the current flowing through the battery.

The pin GND1 (JP6) is a reference for the ground level on the UEPM Development Board.

The connector JP5 is connected in series with the battery in order to control the current flowing through it, with the help of an amperemeter. If not in use, place a jumper on JP5.



Figure 7 – Control Pins (on the Left)

CAUTION!

Do not short JP5 and JP6 as you will create a short circuit and you may permanently damage the battery and other components of the UEPM Development Board.

4.6. Switches

The UEPM Development Board is equipped with three switches:

- SW1: main switch. It connects the battery to the input of the regulator. SW1 is bright red when closed.
- SW2 and SW3: enabling switches. They enable the regulator to let the current flow to power the UEPM Development Board when closed. SW2 and SW3 are in parallel. SW2 is SPST ON-OFF. SW3 is a momentary SPDT ON-OFF. SW2 is bright green when SW2 or SW3 is closed.

When the voltage regulator of the UEPM Development Board is powered and enabled (SW1 & (SW2+SW3)) the LED L2 is bright green.

Moreover, the digital output EN_REG2 of the MSP430 is connected to the enable pin of the voltage regulator. Hence, after a short impulse on SW3 to allow current flowing through the voltage regulator and power the MSP430, the latter can set its output EN_REG2 high to always enable the regulator. Only clearing EN_REG2 programmatically or switching off SW1 will shut down the board. The advantage of this configuration is that the program of the MSP430 can control when it needs to totally shut down.



Figure 8 – SW1 and SW2

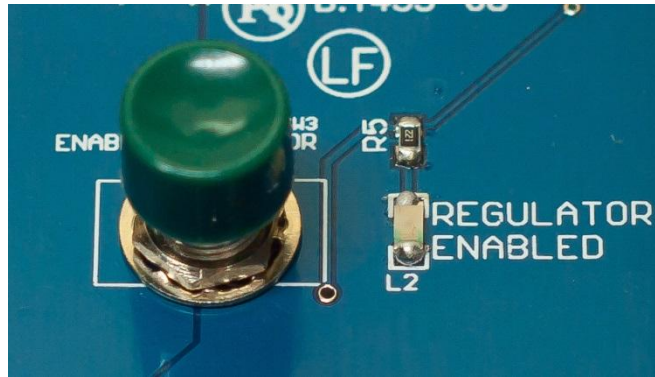


Figure 9 – SW3 and L2

4.7. Serial Communication

The UEPM Development Board uses two UART ports of the MSP430 and one SPI port.

The two UART ports of the MSP430 are USC10 and USC11 and are respectively called on the UEPM Development Board UART1 and UART2.

The SPI port is directly connected to the micro-SD Slot. An additional interruptible input SD_CD is used to detect when a micro-SD card is inserted into the slot.

The UART1 can be used with any XBee footprint compatible module, an INNOVASUB pressure sensor or an underwater communication module (provided optionally). However, only 1 device can be used at the same time. The Jumper JP9 is used to select which device is to be used with UART1.

To use XBEE1 with UART1, REMOVE FIRST the jumpers from the positions UART1 and SNS1, and finally place three jumpers on XBEE1 on JP9. In order to use the INNOVASUB pressure sensor 1, remove first all the jumpers from JP9 and place two jumpers on the position SNS1. In order to use the underwater communication module, remove all the jumpers from JP9 and place three jumpers on the position UART1.

When using XBEE1, CTS and RTS pins are directly wired to the MSP430 if high baud rates needed.

The UART2 can be used with a Xbee footprint compatible module or an INNOVASUB pressure sensor. However, only 1 device can be used at the same time. The Jumper JP10 is used to select which device to use with UART2.

To use XBEE2 with UART2, REMOVE FIRST the jumpers from the position SNS2, and finally place three jumpers on XBEE2 on JP10. In order to use the INNOVASUB pressure sensor 2, remove first all the jumpers from JP10 and place two jumpers on the position SNS2.

When using XBEE2, its CTS and RTS pins are directly wired to the MSP430 in case high baud rates needed.

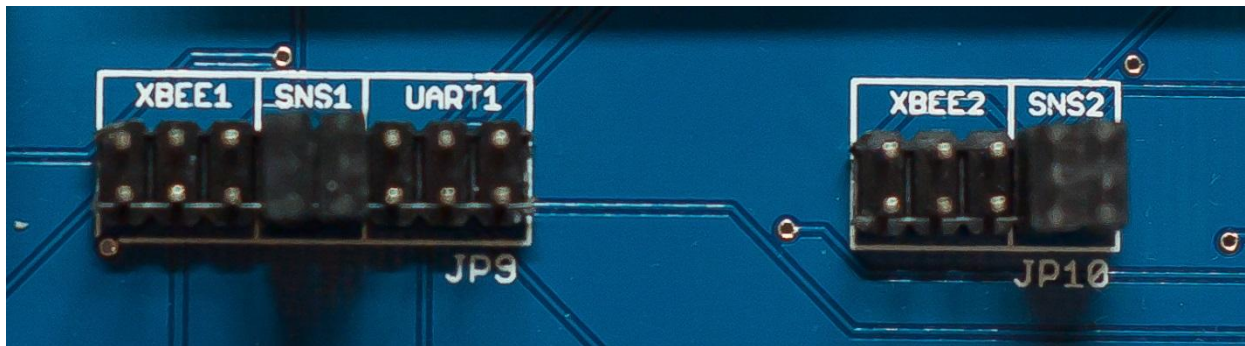


Figure 10 – Jumper JP9 in Position SNS1 and Jumper JP10 in Position SNS2

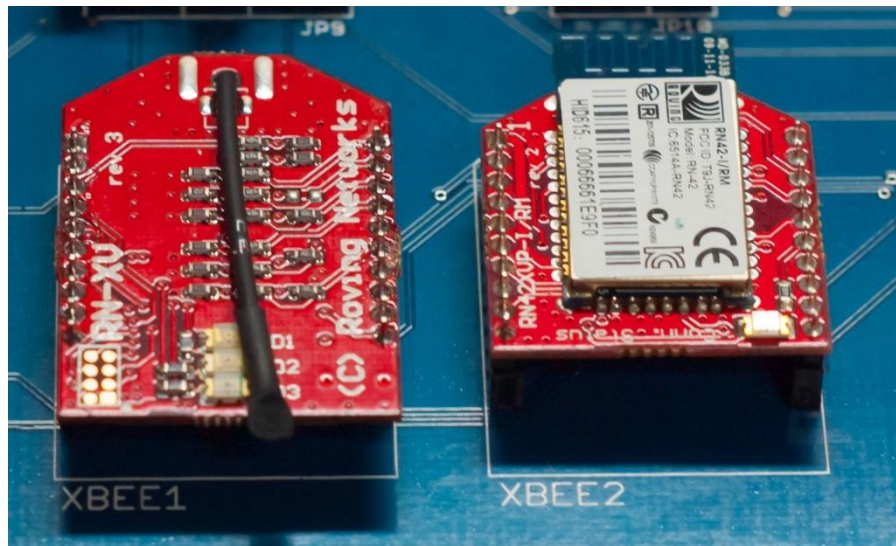


Figure 11 – XBEE1 (RN-171XV) and XBEE2 (RN-42XV)

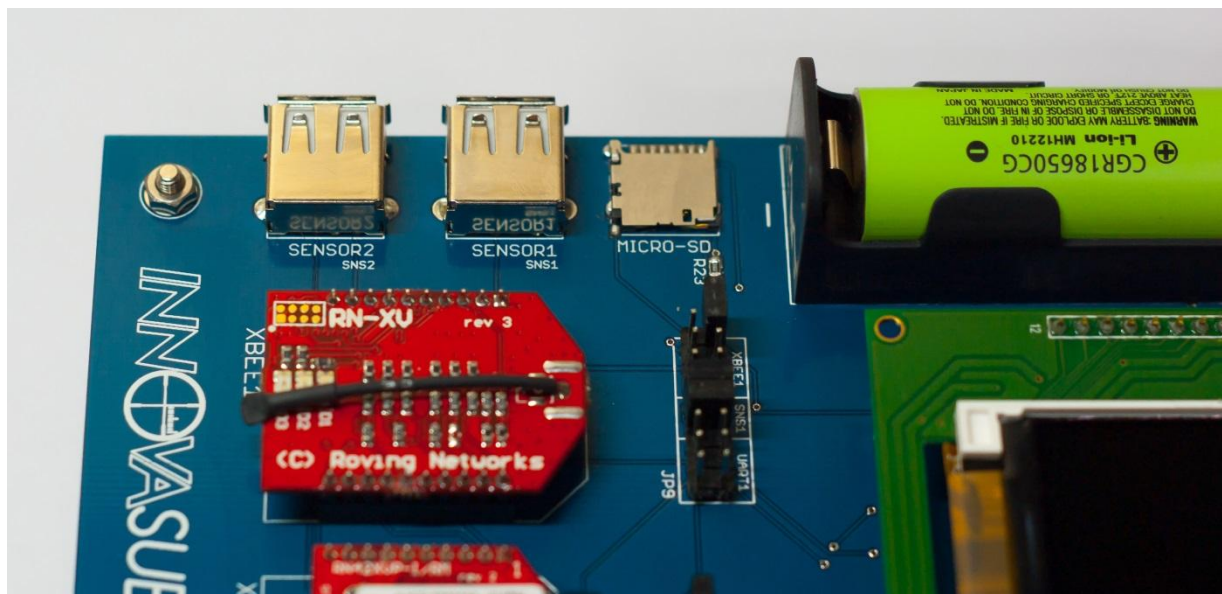


Figure 12 – SENSOR2, SENSOR1 and Micro-SD Slot

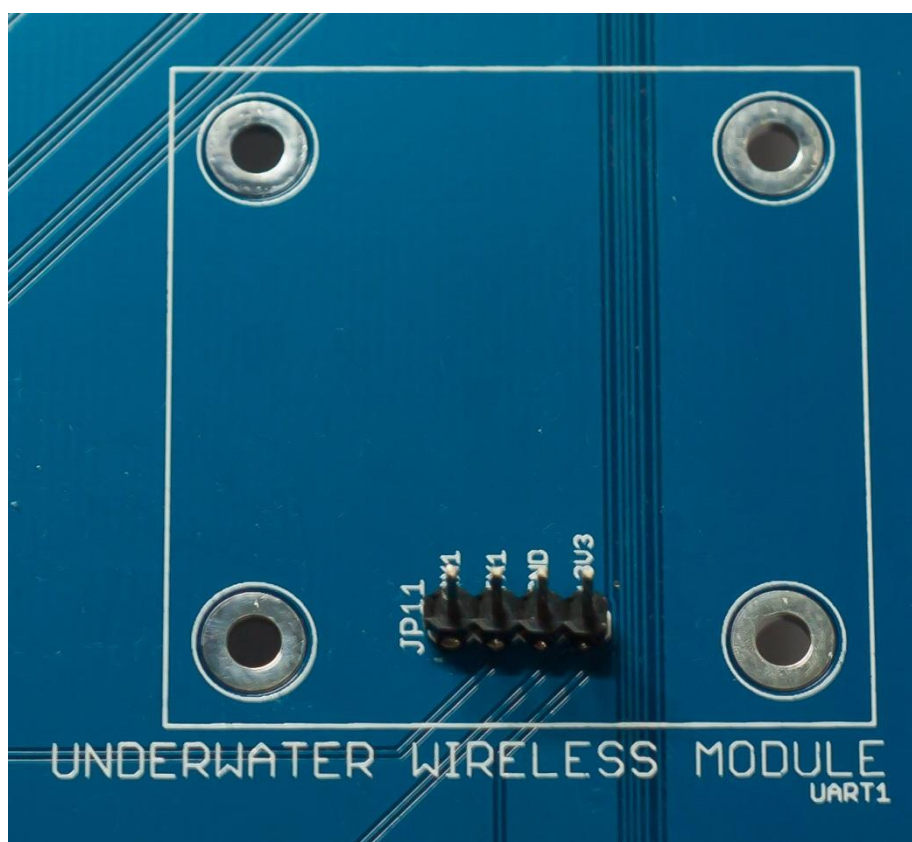


Figure 13 – Underwater Wireless Module Connector

CAUTION!

SENSOR1 and SENSOR2 connectors are NOT USB COMPLIANT. Do not connect any other device than the INNOVASUB Pressure Sensors.

4.8. INNOVASUB Pressure Sensor

The INNOVASUB Pressure Sensor is a high precision pressure sensor to monitor your divers' depths. Based on the MS5541C, its resolution is 1.2 mbar and its range is up to 14 bar.

The INNOVASUB Pressure Sensor easily plugs to your existing panel system using the T-pipes adding digital sensing for your operations.



Figure 14 – Two INNOVASUB Pressure Sensors and Two T-Pipes



Figure 15 –INNOVASUB Pressure Sensors with USB Cable for Connection with the UEPM Development Board

4.9. Piezo-Switches

Piezo-Switches are often used on dive computers and other underwater electronic devices that require switches as they are completely water proof and they easily work underwater. A small impulse of the piezo-electric switch will close the circuit for a short period of time which will be detected by the MSP430's input.

PIEZO1 is wired to the input P2.2 and PIEZO2 is wired to the input P2.3.



Figure 16 – PIEZO1 and PIEZO2 Piezo-Switches

4.10. Potentiometers



Figure 17 – POT1 and POT2 Potentiometers

POT1 and POT2 are high precision, 10-turn potentiometers with knob. They are respectively wired to the A2 and A3 12-bit ADC inputs of the MSP430.

Notice: The signals of the potentiometers are inverted. The position 0.0 corresponds to the highest signal in ADC acquisition and 10.0 to the smallest value.

4.11. LEDs

Four general purpose LEDs are available on the UEPM Development Board. They can be used to display basic information. They are called LED1, LED2, LED3 and LED4 and their signals are of the same names on the circuit's schematics. They are directly connected to the MSP430 digital outputs and they use 220 Ohm resistors to limit current.

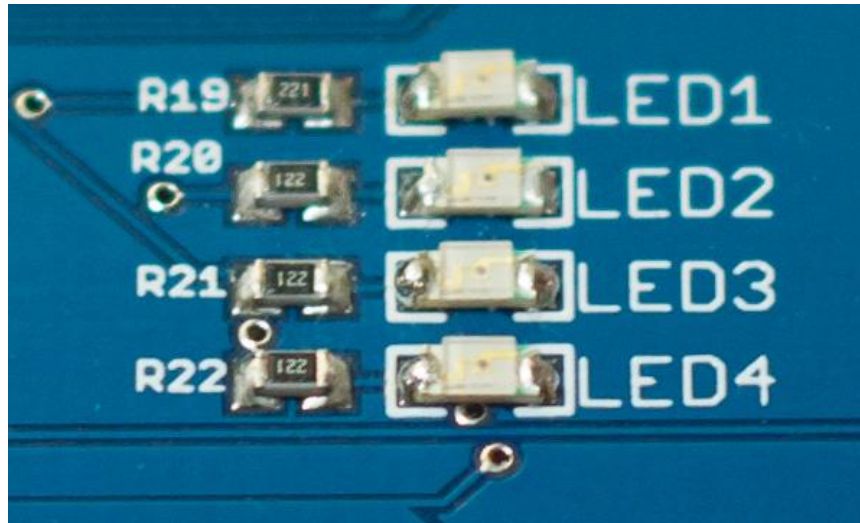


Figure 18 – LED1, LED2, LED3 and LED4

4.12. Oxygen Sensor

The connector X4 (jack 3.5mm) and JP15 (0.1" pitch standard) are designed to be connected to an analog oxygen sensor. They are connected to the 12-bit ADC input A4.

CAUTION!

Do not connect both X4 and JP15 to an oxygen sensor at the same time as they are wired to the same input and may result to damage the two sensors.

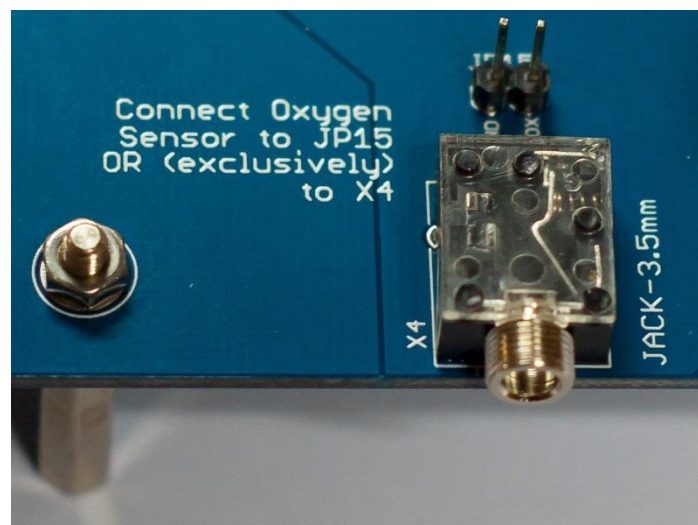


Figure 19 – X4 (Jack 3.5mm) and JP15 (0.1") Connectors for Oxygen Sensor

4.13. General Purpose ADC Inputs

The connector JP16 offers to connect a custom general purpose board for up to 6 12-bit Analog-To-Digital inputs. It also provides the battery lines, the regulated 3.3V, the analog ground (AGND) and digital ground (DGND) for a high quality signal.

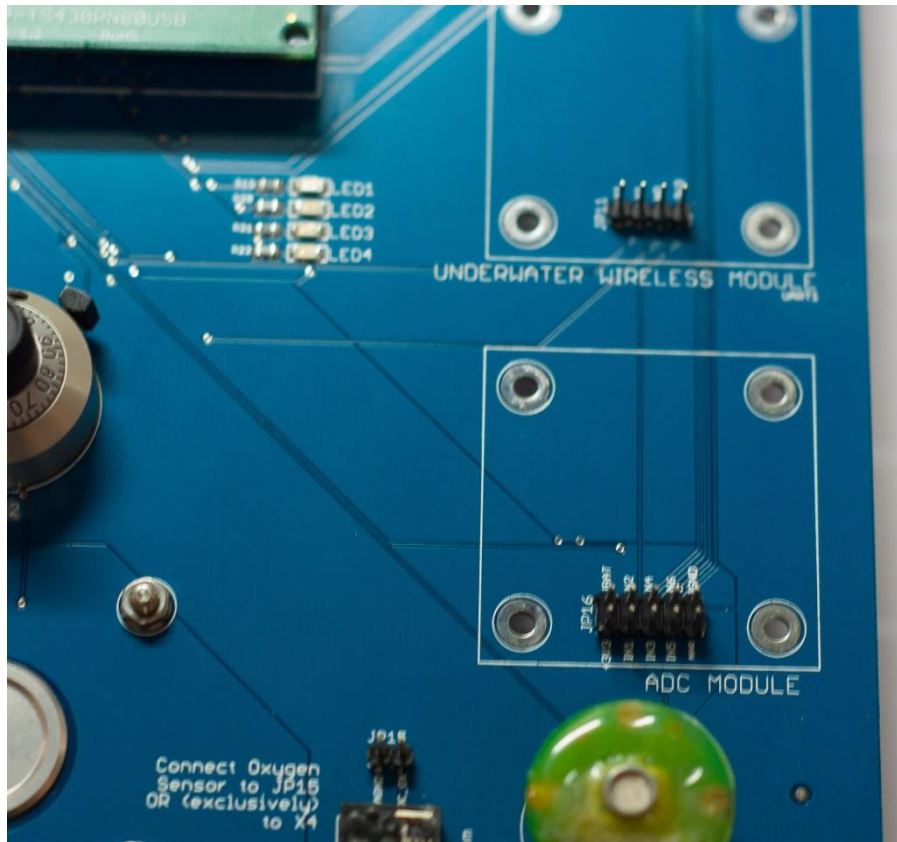


Figure 20 – ADC Module – JP16 Connector for 6 x 12-bit ADC Inputs

4.14. MS5541C Pressure Sensor

The MS5541C is mounted on an INNOVASUB PCB for easy experimenting and development of the communication protocol with the pressure sensor. The connection is made with 5 pins only.

CAUTION!

Do not invert the sensor's connection while plugging it. It may result in damaging it. Follow the PCB's drawing to plug it in the correct direction.

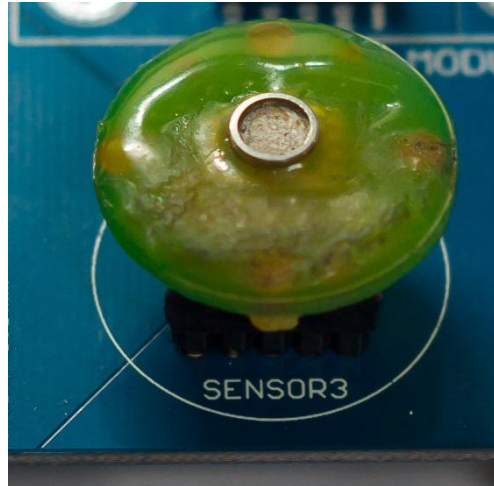


Figure 21 – SENSOR3

5. PROGRAMS

5.1. Sample Codes

The UEPM Development Board is provided with a sample code to illustrate how to use each component. Three sample programs are provided.

Program	Description
USB Program	Contains the basic structure for using USB with MSP430. Illustrates the exchange of data via a communication between the MSP430 and a terminal application on PC through USB.
SD Card Program	Provided by Texas Instruments. Sample code to illustrate how to read and store data on an SD memory card. IT DOES NOT implement FAT file system.
UEPM Sample Program	Illustrates all other features offered by the UEPM Development board. Namely: auto-shutting down, circuit's and battery's voltages monitoring, serial communication through XBee, INNOVASUB Pressure sensor, MS5541C pressure sensor, high precision 10-turn potentiometers, piezo-switches, LEDs display and LCD Display.

The USB program was set apart because it requires a specific structure in the main loop (main.c) that would have made the rest of the code harder to understand at first. The SD Card program was also set apart because it requires a long code that may confuse the developer when first using the UEPM Development Board. Both the USB and SD card programs were directly inspired from sample codes provided by Texas Instruments.

The UEPM Sample program was developed by INNOVASUB and it illustrates all other functions of the UEPM Development Board.

All the code was developed in the Code Composer Studio environment. Import the program to CCS as a project. You may be required to change the includes of the C program since the environment might be different.

5.2. USB Sample Code

The USB Sample Code provided with the UEPM Development Board was based on the Texas Instruments USB API and sample codes and was developed by INNOVASUB. It illustrates a simple USB CDC interface between the MSP430 and a computer's terminal

application. The USB MSP430 requires the user to enter a password on the terminal application. Then, the MSP430 informs the user whether the password is correct or not.

To execute this sample,

- First install a terminal application on your computer. If you do not have any, install `teraterm-4.79.exe` provided for Windows operating systems with the UEPM Development Board's CD.
- Open the project UEPM-USB in code composer studio. Arrange the includes if necessary.
- Load and run the program to the MSP430F5529 Microcontroller.
- Connect a USB cable between a computer and the MSP430 target board.
- On Windows, if required, manually point the driver installer to the file `MSP430_CDC.inf`. Make sure the driver is installed properly.
- Open a terminal application such as Tera Term. Open CDC COMM PORT of the USB.
- In the terminal, type any key. Then enter the password: "innovasub".

For further information on how to use the Texas Instruments USB API, please refer to the document `Programmers_Guide_MSP430_USB_API.pdf` provided with the UEPM Development Board's CD.

5.3. SD Card Sample

The SD Card slot is connected to the MSP430 through its SPI port. The library for SD implementation, a sample code and the datasheet are provided in the `/SD/` folder of the CD.

Please notice that this library only implements byte reading and writing. It DOES NOT IMPLEMENT FAT FILE SYSTEM. Therefore, the data written on the SD card will not be directly readable on any operating system.

To implement a FAT file system, you are strongly advised to use the open source library FatFs available here: http://elm-chan.org/fsw/ff/00index_e.html

5.4. UEPM Sample Code

The main function of the sample code is contained in `main.c`. Each peripheral of the UEPM Development Board has its own source and header files. Namely:

- `adc`: manages the Analog-to-Digital Conversion of the 12-bit ADC.
- `interrupt`: gathers all the interrupts of the program. Digital Input interrupts, ADC acquisition interrupts, Timer A interrupts and UART interrupts.
- `lcd`: manages the LCD display.
- `led`: manages the four LEDs of the UEPM Development Board.

- piezo: manages the piezo-switches input.
- powermanager: manages the Enable Pin of the regulator of the board. Can decide when to disable the enable pin and therefore shut the board down.
- pressuresensor: manages the communication with SENSOR3 (MS541C).
- sdsdslot: detects when an SD card was inserted to the slot. Do not interface with the SD card itself. Please see the *5.3 SD Card Sample* for interfacing with the SD card.
- timer: configures the clocks and timers of the board for a proper time management.
- uart: configures and implements serial communication.
- Xbee: configures and manages the pins specific to the XBee port (status pins).

Additionally, the following files are comprised in the project:

- utils.c and utils.h: contains functions that can be used across the whole project.
- configuration.h: constants used for pre-compiling the project.
- mapping.h: defines the position of each pins function. For example, LED1 digital output is placed in P7.3. This header file ensures the compatibility of all the source codes mentioned above across all MSP430.

Configuration

Configuration.h contains pre-compiler instructions and definitions.

The only variables that you may want to change there are UART1_FCT and UART2_FCT. They select which device will be used for UART1 and UART2. For example, if you are going to use the XBee for UART1 and the INNOVASUB Pressure Sensor for UART2, then define:

```
#define UART1_FCT XBEE1
#define UART2_FCT SENSOR2
```

Mapping

Mapping.h defines the port and byte number of the digital input/output that are used on the MSP430F5529. Mapping.h ensures that the other files are multiplatform. For example, if you change the circuit for your product (changing the connections) you may only need to modify mapping.h.

Main

Main.h contains the main function. It starts with stopping the watch dog. Then it initializes the clock so that the rest of the program is executed in a controlled time frame. Then it initializes the rest of the peripherals.

Finally, it allows interrupts and enters the main loop. At the end of each execution of the main loop, the program enters the Low-Power Mode 0. Only the interrupts force to leave the LPM0.

The main loop contains a routine for the pressure sensor because it regularly needs to be checked its status. If the pressure does not answer (for example if it was removed and placed again), it is reset.

As a demonstration code, the main.c gives access to the following global variables:

- switch1 and switch2: 1 byte numbers counting the number of impulses on SW1 PIEZO1 and PIEZO2.
- adc12_3V3: adc value of the 3.3V line.
- adc12_bat: adc value of the battery level.
- adc12_pot1 and adc12_pot2: adc values of the POT1 and POT2.
- sensor1..3: pressure value in mbar of SENSOR1, SENSOR2 and SENSOR3 when available.
- sensTp1..3: temperature value in °C of SENSOR1, SENSOR2 and SENSOR3 when available.

Timer

The timer.c and timer.h only contain two initialization functions that configure the TimerA to count timer based on the XT1 32,768 Hz. It generates 8 TimerA overflow interrupts per second that are handled in interrupt.c.

The second initialization function sets the DCO to a precise 19,988,840 Hz. MCLK is set to 19,988,840 Hz and SMCLK is set to MCLK/2. This is the function to be modified if you need to change the MSP430's frequency.

LEDs

The led source and header files contain simple functions to set and clear the LED 1 to 4. They allow to move up or down their positions or to display a lighting effect for start-up.

PowerManager

The power manager sets and clears the enable pin EN_REG2 for the main regulator of the board. At start-up, the power manager sets EN_REG2 high. If EN_REG1 goes low after start-up, then the power manager controls when the whole board will shut down.

Under certain conditions, the power manager can decide to clear EN_REG2 which will dis-allow the main regulator and shut down the board. The only way to start the board again is to push SW2 or SW3 to set EN_REG1 high again and power the MSP430.

Please note in the sample code, the MSP430 set EN_REG2 high at start-up but never clears it. Therefore, to shut down the UEPM Development Board, you need to clear SW1 off.

LCD

The LCD code is not implemented in the UEPM sample code.

UART

The UART source code configures the UART ports according to the configuration. Please note that the XBees are configured to a baud rate of 2400 whereas the INNOVASUB Pressure Sensors are configured for a baud rate of 9600.

UART source code also implements the low level serial communication buffering for a highly efficient communication. The data to be transmitted or received is buffered and called by interrupts only.

XBee

The XBee source code implements the status pins of the XBee modules. These status pins inform on the connection status of the module (TCP/IP and WiFi/Bluetooth association).

Please note that the RTS/CTS function is not implemented. It is advised that they be implemented in this file if required.

Piezo

The piezo source code contains the function to initialize the inputs for the piezo-switches. It setups an interrupt to be called at every piezo-switch impulse. The interrupt increments a counter to count the number of impulses of the each switch and it moves the LEDs up when PIEZO1 is pressed and down when PIEZO2 is pressed.

SD Slot

The SD Slot source code setups an interruptible input to detect when a Micro-SD card is inserted to the Micro-SD Slot.

Note: this is not implementing the read/write functions to the card. But it would be advised to implement the read/write functions (if needed) in this part of the project.

ADC

The ADC source code configures the ADC inputs and the ADC interrupt. In the sample code, it implements A0, A1, A2 and A3 ADC acquisitions in a single-sequence mode and triggers an interrupt when A3 conversion is over.

The interrupt reads the values of the four inputs at the same time. If the conversion is over, the TimerA interrupt triggers a new single-sequence conversion every second.

If you need to add inputs for ADC, you are recommended to modify configuration in `adc.c` initialization function.

Pressure Sensor

Pressure Sensor source and header files implements the non-standard communication between the MSP430 and the MS5541C. In order to make communication as efficient as possible, the communication implements usage of interrupts. It regularly checks the status of the pressure sensor and reset it if it cannot communication with it after 3 seconds.

The Pressure Sensor source code can be used as a library for MS5541C for any other project.

Interrupts

The interrupt source code gathers all the interrupts in a single file. It contains the TimerA overflow (executed 8 times a second), the ADC interrupt (executed once after a sequence of conversion), the UART1 and UART2 interrupts (executed every time a byte is received or sent) and the PORT1 and PORT2 input interrupts.