# Mercury System BB110 – Base Board Model A

The BB110 is a Mercury System Base Board (BB). The BB family represent the Mercury System’s main logical unit and stores the user application. This board interacts with several Slave Boards (SBs) and Modem Boards (MBs) in order to implements various connected and IoT applications. The heart of the system is a PIC18F46J50 8-bit RISC microcontroller, produced by Microchip Technology Inc.

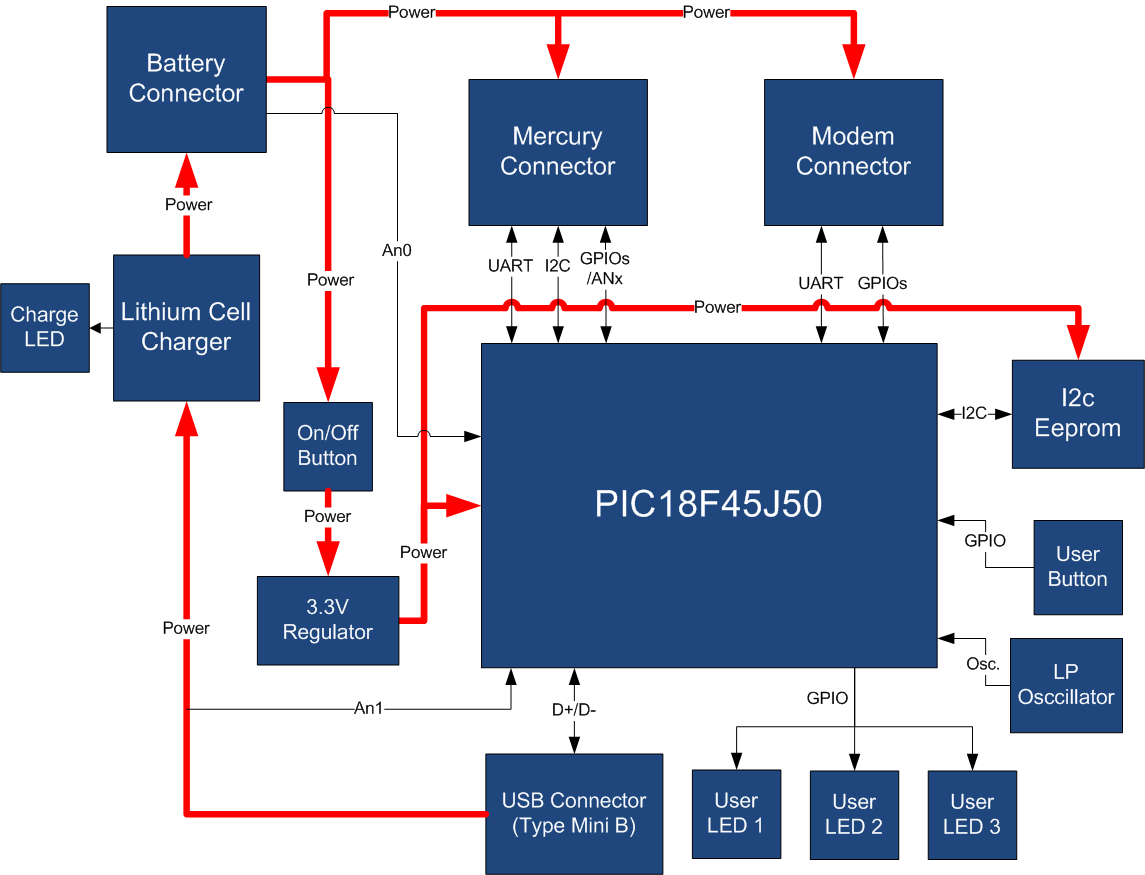


Figure 1 - BB110 Block Diagram

The main characteristics of the employed MCU are resumed in Table 1:

Table 1 - MCU characteristics

|  |  |
| --- | --- |
| Parameter Name | Description |
| Program Memory Type | Flash |
| Program Memory (KB) | 64 |
| CPU Speed (MIPS) | 12 |
| RAM Bytes | 3,800 |
| Digital Communication Peripherals | 2-UART, 2-A/E/USART, 2-SPI, 2-I2C2-MSSP(SPI/I2C) |
| Capture/Compare/PWM Peripherals | 2 ECCP |
| Timers | 2 x 8-bit, 3 x 16-bit |
| ADC | 13 ch, 10-bit |
| Comparators | 2 |
| USB (ch, speed, compliance) | 1, FS Device, USB 2.0 |
| Temperature Range (C) | -40 to 85 |
| Operating Voltage Range (V) | 2 to 3.6 |
| Pin Count | 44 |
| XLP | Yes |
| Cap Touch Channels | 13 |

The board is equipped with two connectors, the Mercury connector and the Modem connector, so it can be interfaced with Mercury System Slave boards (SB) and Mercury System Modem Board (MB). Moreover the board is equipped with a USB connector, for PC connectivity and bootloading purposes. The board features also an I2C EEPROM for non-volatile data storage, LowPower oscillator to enable MCU’s RTCC, some user LEDs and one user button. To satisfy power requirements of simple systems the board has an integrated 3,3V voltage regulator able to supply up to 1A and a single cell LiPo recharging circuitry.

Table 2 resumes the BB110 board main characteristics:

Table 2 - BB110 Characteristics

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Notes |
| MCU Family | PIC18F |  |
| Supported Buses | I2C, UART |  |
| Connectivity | USB device |  |
| Eeprom Size | 2Kbyte |  |
| Max Reg. Power | 1A |  |
| Total IO number | 11 |  |
| RTCC | Yes |  |
| LiPo Battery Rech. Circuit | Yes |  |

Figure 2 depicts the most important components of the board:

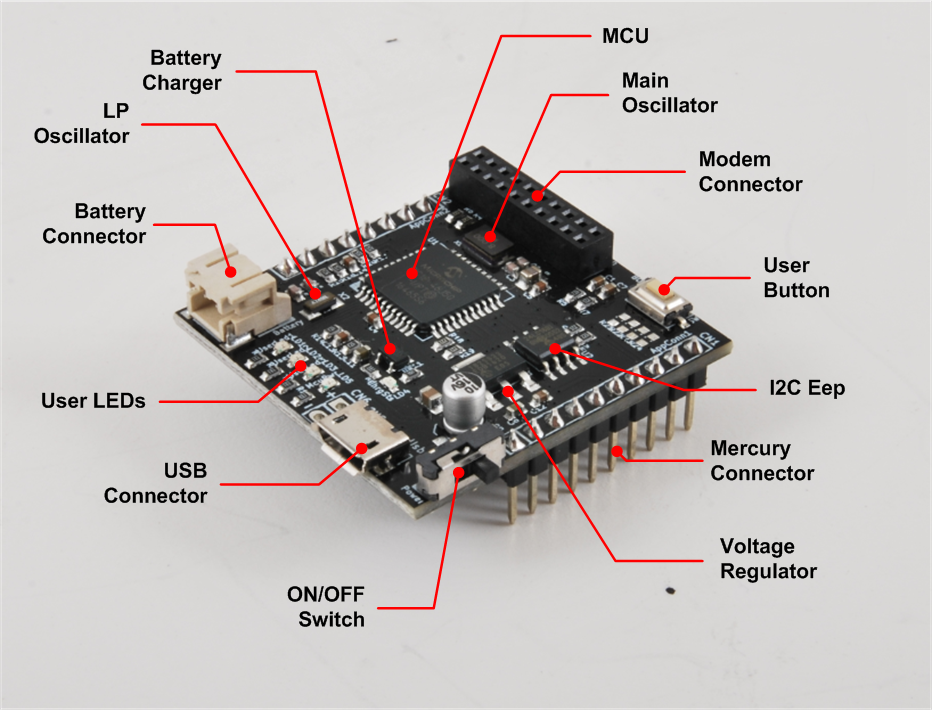


Figure 2 - BB110 Hardware Highlight

Table 3 provides a description of board’s main components:

Table 3 – BB110 Hardware characteristics

|  |  |
| --- | --- |
| Parameter Name | Description |
| User LEDs | 3 General Purpose User LEDs. |
| USB Connector | Connector of Full-Speed USB 2.0 bus. |
| ON/OFF Switch | Main Power Switch. |
| Voltage Regulator | 3,3V x 1A Voltage Regulator |
| Mercury Connector | Mercury Connector for the interfacing of Mercury Slaves. |
| I2C Eep | On-board I2C 16kbit EEPROM. |
| User Button | General Purpose User Button. |
| Modem Connector | Mercury Modem Connector for interfacing Mercury Modem Boards. |
| Main Oscillator | 8MHz main oscillator. |
| MCU | PIC18F46J50 Main Microcontroller Unit. |
| Battery Charger | Single cell LiPo battery charger. |
| LP Oscillator | Low Power 32KHz oscillator for internal RTCC. |
| Battery Connector | Connector for external LiPo battery. |

# Mercury System Framework

The Mercury System Framework (MSF) is a layered SW framework specifically designed to support application development with Mercury System. It provides to the user a complete set of base functionalities to easily interface MS Slave Boards (SB) and Modem Boards (MB) as well as some infrastructural and system services SW. Figure 3 shows the layered Architecture of MSF.

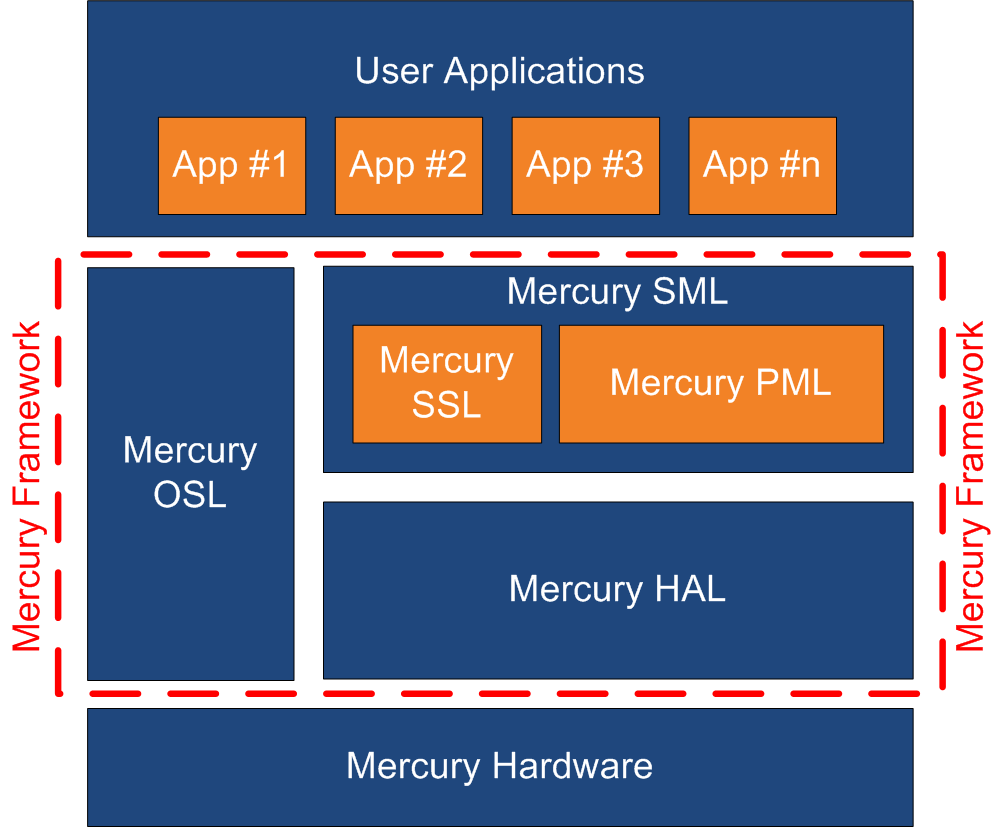


Figure 3 - Mercury System Framework Architecture

The framework is made up by the following components:

**HAL (Hardware Abstraction Layer):** the purpose of this layer is to abstract the HW dependencies to the upper layers.

**SML (System Management Layer):** the purpose of this layer is to provide services for the management of communication buses (I2C, UART) and for the management of Mercury System’s Modem Board (WiFi, BT, GSM/GPRS). It also provides a set of System Services, like System Power Management, RTCC, USB terminal, etc. It’s divided in two main components:

* PML: Peripheral Management Layer,
* SSL: System Services Layer.

**OSL (Operative System Layer):** this layer is made up by a lightweight RTOS that provides basic services to the system, like scheduling tables for the various tasks, Events, SW Timers, Alarms, etc…

Figure 4 provides a symbolic example about the positioning of a user application inside the MSF layered architecture.

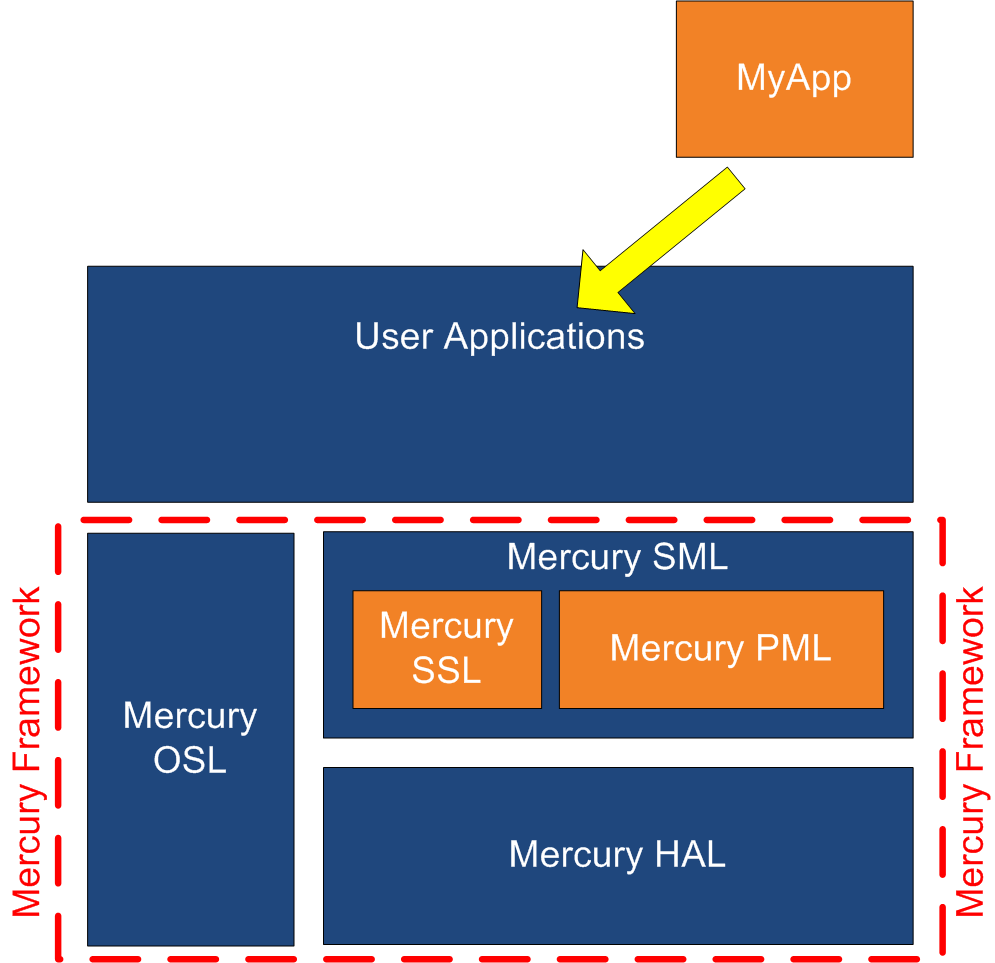


Figure 4 - App development with MSF