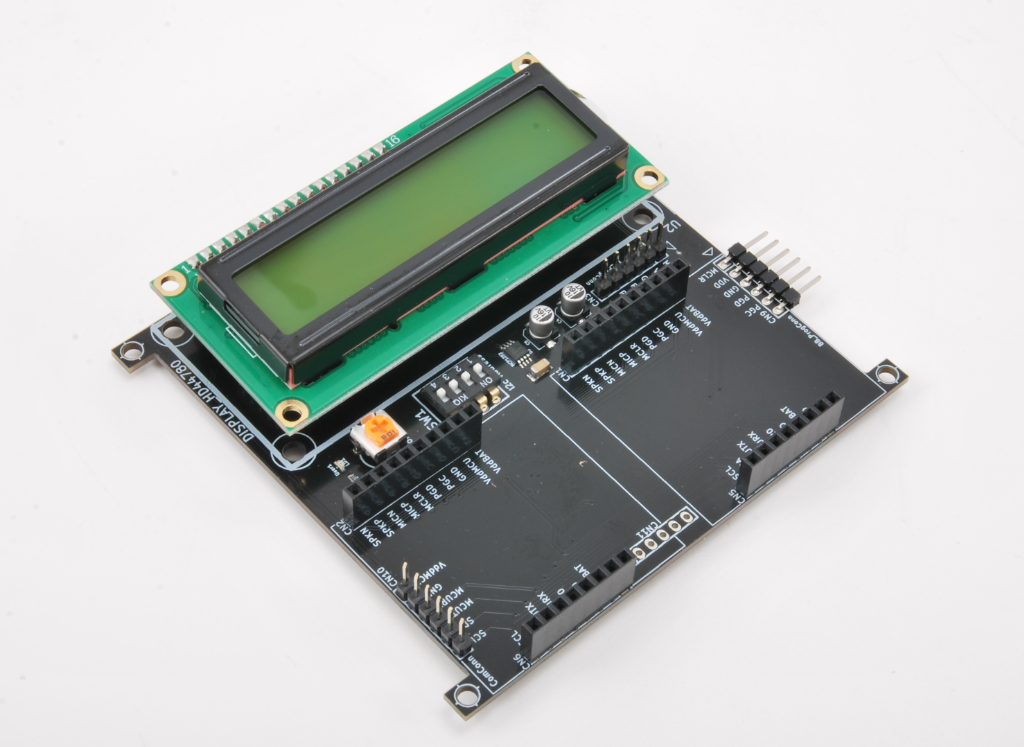
Mercury System

EB210



Expansion Display 16x2 Board - Product Datasheet

|  |  |
| --- | --- |
| Author | Francesco Ficili |
| Date | 01/11/2018 |
| Status | Released |

|  |  |  |  |
| --- | --- | --- | --- |
| Revision History | | | |
| Version | Date | Author | Changes |
| 1.0 | 28/10/2018 | Francesco Ficili | Initial Release. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**SUMMARY**

[1. Introduction 4](#_Toc528832461)

[2. Block Diagram 6](#_Toc528832462)

[3. Hardware 8](#_Toc528832463)

[4. Pinouts 9](#_Toc528832464)

[Mercury Connector 9](#_Toc528832465)

[EB Programmer Connector 10](#_Toc528832466)

[BB Programmer Connector 11](#_Toc528832467)

[Communication Connector 12](#_Toc528832468)

[Expansion Connector 12](#_Toc528832469)

[5. Command Set 14](#_Toc528832470)

[Specific Command Set 14](#_Toc528832471)

[Examples 14](#_Toc528832472)

[6. Technical Specifications 15](#_Toc528832473)

# Introduction

The Mercury System (MS in short) is a modular system for the development of connectivity and IoT applications. The system uses various type of electronic boards (logic unit, modems, slave board equipped with sensors and actuators, power boards...) and a complete SW framework to allow the realization of complex applications. Scalability, ease of use and modularity are key factors and are granted by the use of a heterogeneous set of components that allow to assemble the system like a construction made with LEGO© bricks.

The board set which composes the system is made up by the following “families”:

* **Base Board (BB):** It’s the “brain” of the system and contains the main logic unit as well as different communication buses and connector to interfaces the slaves. It also contains a simple power supply system and a recharge unit for a single LiPo cell (it can satisfy the power requirements of simpler systems). It can exist in different variants, depending on the employed microcontroller unit.
* **Modem Board (MB):** this one is the board that allow network connectivity. It can exist in different variant, depending on the network interface (GSM/GPRS, Wi-Fi, BT, Radio…). It’s interfaced to the Base Board with a dedicated serial line.
* **Power Board (PB):** it’s the board that allow to satisfy the particular power requirement of the system, when it’s necessary. They can be vary depending on the particular power requirement to satisfy (high power, solar harvesting, piezo harvesting, etc.).
* **Slave Board (SB):** these are the system’s peripherals, and they vary depending on the specific mounted sensor or actuator. Typical examples are SB with relay, temperature sensors, RGB LED controller, servo controller, accelerometer, etc. They communicate with the BB with I2C or UART and a dedicated command set.
* **Expansion Board (EB):** these are the board that allow planar connection of Mercury boards. There are variants which can contains Displays, battery socket, etc.
* **Brain-Less Board (BL):** these are the controller-less boards. They in general contain really simple sensor or actuators that don’t need the bus interface. There are meant as an alternative to slave boards for cost-sensitive applications.

Slave Boards and Modem Board are provided pre-programmed with a FW which implements a dedicated command set for a high-level management of the boards, while the Base Boards are provided with a SW framework which provides all the low-level services (operative system, device drivers, system services, etc.), leaving to the user only the development of application level logic. Moreover, the Base Board comes with an USB bootloader, so it can be programmed without the need of a flashing device.

Figure 1 shows a typical system connection:

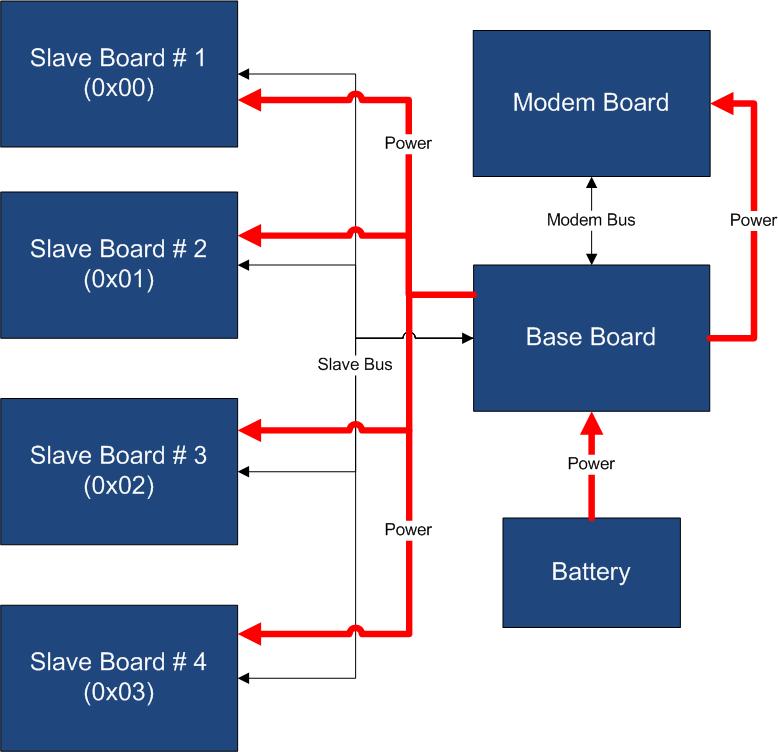


Figure 1 - Typical System Connection

Examples of application fields of MS are:

* Home automation System,
* IoT applications,
* Connectivity Applications,
* Monitoring and control Systems,
* Remote Control,
* Industrial Process control,
* Robotics applications,
* Test benches,
* Etc…

# Block Diagram

The EB210 is a Mercury System expansion board with integrated alphanumeric 16x2 display. The board allows the classic planar expansion offered by Mercury System expansion board, and at the same time provides a way to build a simple User Interface. Figure 2 shows the EB210 block diagram. The heart of the system is a PIC16F1829 8-bit RISC microcontroller, produced by Microchip Technology Inc.

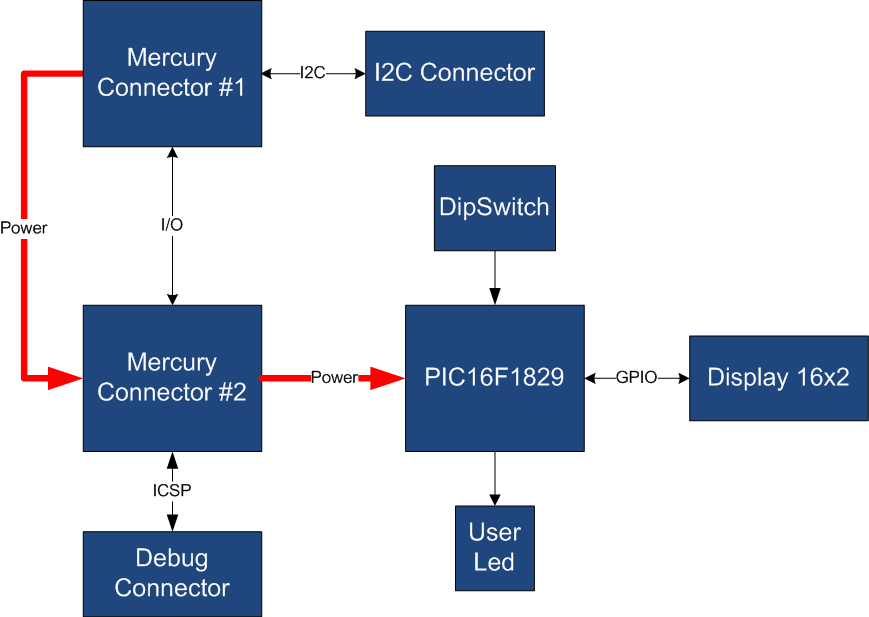


Figure 2 - Block Diagram

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) | 14 |
| CPU Speed (MIPS) | 8 |
| RAM Bytes | 1,024 |
| Data EEPROM (bytes) | 256 |
| Digital Communication Peripherals | 1-UART, 1-A/E/USART, 1-SPI, 1-I2C1-MSSP(SPI/I2C) |
| Capture/Compare/PWM Peripherals | 2 CCP, 2 ECCP |
| Timers | 4 x 8-bit, 1 x 16-bit |
| ADC | 12 ch, 10-bit |
| Comparators | 2 |
| Temperature Range (C) | -40 to 125 |
| Operating Voltage Range (V) | 1.8 to 5.5 |
| Pin Count | 20 |
| XLP | Yes |

The EB210 is connected to the BB by means of I2C bus. The address of the board could be dynamically set by means of a 4 positions dip switch, allowing up to 15 address values (address 0x00 is reserved for I2C general call broadcast addressing scheme).

Table 2 resumes the EB210 board main characteristics:

Table 2 – Board Characteristics

|  |  |  |
| --- | --- | --- |
| Parameter | Description | Notes |
| Board Type | Expansion Board (EB) |  |
| Supported Bus | I2C |  |
| Addressing | Dip Switch 4 |  |
| Peripheral Description | Alphanumeric 16x2 LCD Display |  |

# Hardware

This section goes deeper in the HW details of EB210. Figure 3 depicts the most important components of the board:

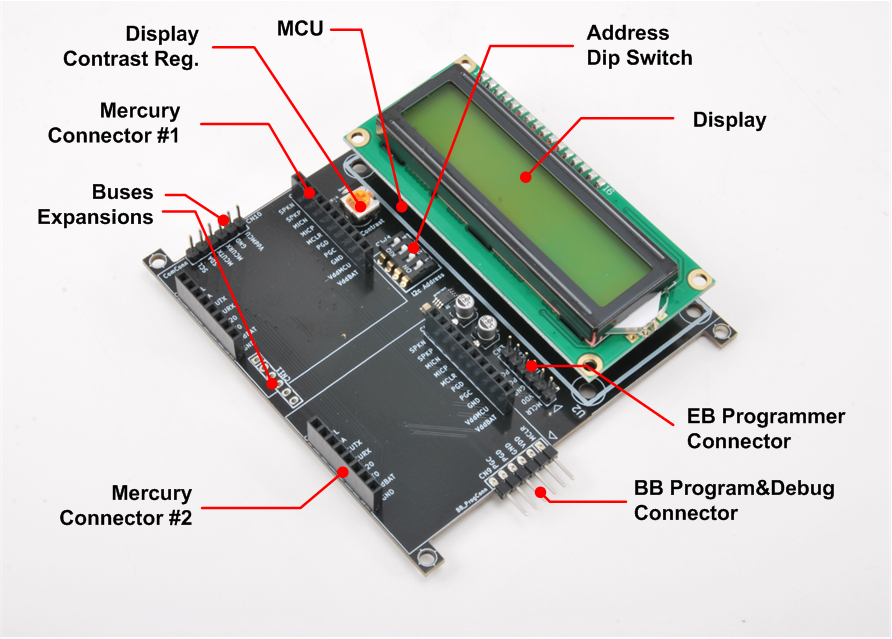


Figure 3 - Hardware Highlight

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

Table 3 – Hardware characteristics

|  |  |
| --- | --- |
| Name | Description |
| Mercury Connectors #1 and #2 | Mercury connector used to interface the board with the others MS boards. |
| Address Dip Switch | Dip Switch to set the address of the board within the Mercury System. |
| MCU | PIC16F1829 main controller board. |
| EB Programmer Connector | PicKit 3 Microchip Programmer/debugger connector. It is directly connected to the EB (Expansion Board) MCU debug port, in order to allow advanced debugging and programming features, if needed. |
| BB Program&Debug Connector | PicKit 3 Microchip Programmer/debugger connector. It is directly connected to the BB (Base Board) MCU debug port, in order to allow advanced debugging and programming features, if needed. |
| Display | 16x2 Alphanumeric LCD Display. |
| Display Contrast Reg. | Potentiometer for display contrast regulation. |
| Buses Expansion | Expansions connectors that carry power as well as I2C and UART connection. |

# Pinouts

This section highlights the pinouts of EB210 connectors.

## Mercury Connector

The Mercury Connector is the connector which interfaces the BB210 with the rest of Mercury System. The connector’s pinout is depicted in Figure 4 and Table 4 explains the meaning of each single pin (NC stands for “Not Connected”).

Table 4 - Mercury Connector Pinout

|  |  |  |
| --- | --- | --- |
| Pin Name | Pin Number | Description |
| VddBat | CN1,2 – 1 | This pin is connected to the main power source. |
| VddMcu | CN1,2 – 2 | This pin is connected to MCU regulated positive voltage reference (3,3V). |
| GND | CN1,2 – 3 | This pin is connected to the board reference voltage. |
| IO11/PGC | CN1,2 – 4 | This pin is connected to BB PGC (Program Clock) line, for advanced debugging features, and can be alternatively be used as generic IO (IO11). |
| IO10/PGD | CN1,2 – 5 | This pin is connected to BB PGD (Program Data) line, for advanced debugging features, and can be alternatively be used as generic IO (IO10). |
| MCLR | CN1,2 – 6 | This pin is connected to BB MCLR (Reset) line. |
| IO9 | CN1,2 – 7 | This pin is connected to BB generic IO6 line. |
| IO8 | CN1,2 – 8 | This pin is connected to BB generic IO7 line. |
| IO7 | CN1,2 – 9 | This pin is connected to BB generic IO8 line. |
| IO6 | CN1,2 – 10 | This pin is connected to BB generic IO9 line. |
| GND | CN5,6 – 1 | This pin is connected to the board reference voltage. |
| VddBat | CN5,6 – 2 | This pin is connected to the main power source. |
| IO5/INT0 | CN5,6 – 3 | This pin is connected to BB INT0 line and can be alternatively be used as generic IO (IO5). |
| IO4/INT1 | CN5,6 – 4 | This pin is connected to BB INT1 line and can be alternatively be used as generic IO (IO4). |
| McuRx/IO3 | CN5,6 – 5 | This pin is connected to BB UART Rx line and can be alternatively be used as generic IO (IO3). |
| McuTx/IO2 | CN5,6 – 6 | This pin is connected to BB UART Tx line and can be alternatively be used as generic IO (IO2). |
| SDA/IO1 | CN5,6 – 7 | This pin is connected to BB I2C SDA line (Data Line) and can be alternatively be used as generic IO (IO1). |
| SCL/IO0 | CN5,6 – 8 | This pin is connected to BB I2C SCL line (Clock Line) and can be alternatively be used as generic IO (IO0). |

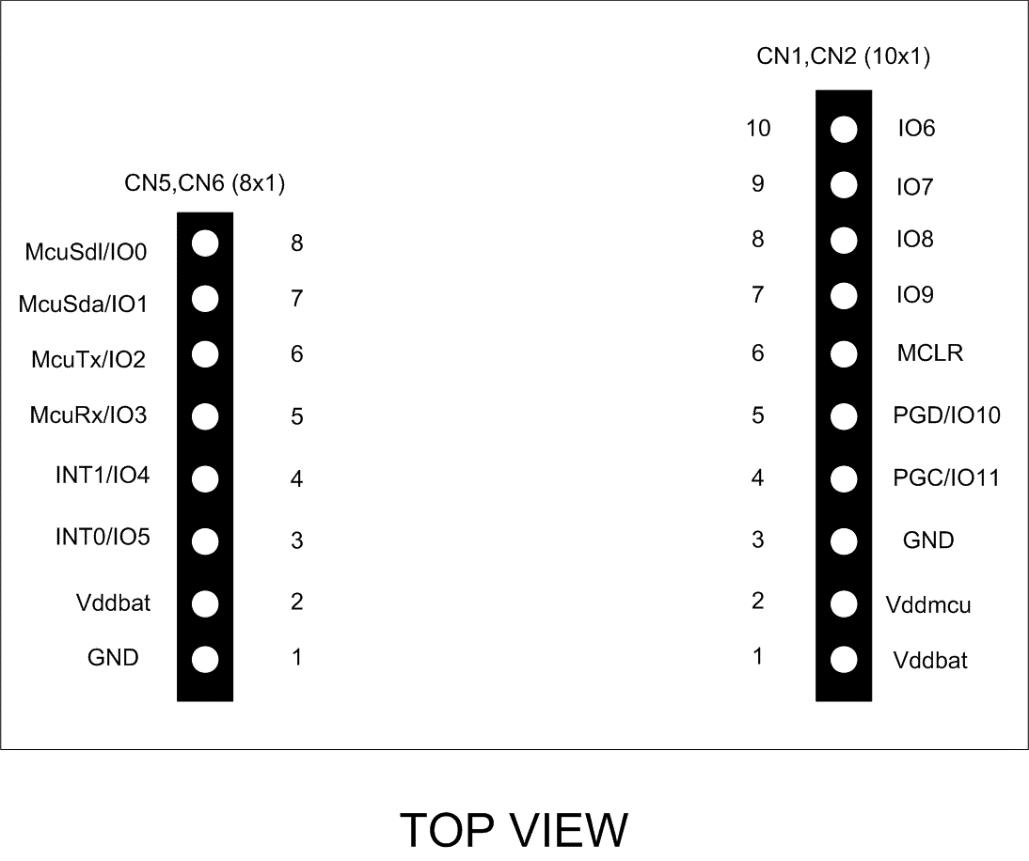


Figure 4 - Mercury Connector Pinout

## EB Programmer Connector

The EB Programmer Connector is the connector which allows to re-program the EB210 using Microchip Technology ICSP (In-Circuit Serial Programming) interface. The connector’s pinout is depicted in Figure 5 and Table 5 explains the meaning of each single pin (NC stands for “Not Connected”).

Table 5 – EB Programmer Connector Pinout

|  |  |  |
| --- | --- | --- |
| Pin Name | Pin Number | Description |
| MCLR | CN3 – 1 | Microcontroller Master Clear (RESET) pin. |
| Vdd | CN3 – 2 | Positive power supply reference. |
| GND | CN3 – 3 | Negative power supply reference. |
| PGD | CN3 – 4 | Program Data pin. |
| PGC | CN3 – 5 | Program Clock pin. |

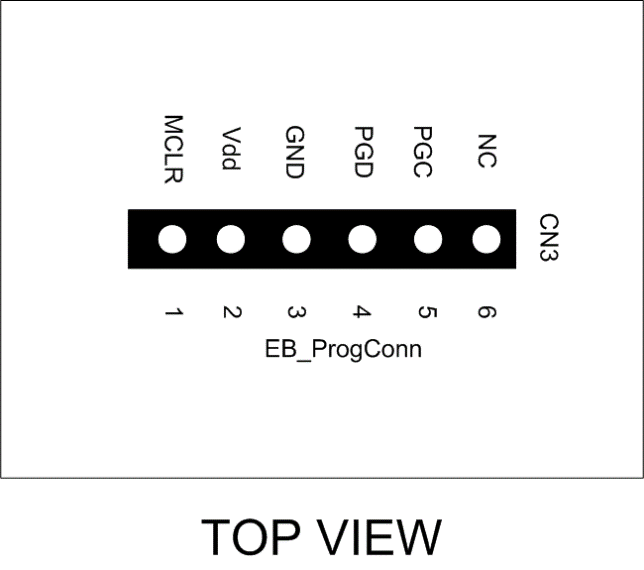


Figure 5 - EB Programmer Connector Pinout

## BB Programmer Connector

The BB Programmer Connector is the connector which allows to re-program the BB (Base Board) using Microchip Technology ICSP (In-Circuit Serial Programming) interface. The connector’s pinout is depicted in Figure 6 and Table 6 explains the meaning of each single pin (NC stands for “Not Connected”).

Table 6 – BB Programmer Connector Pinout

|  |  |  |
| --- | --- | --- |
| Pin Name | Pin Number | Description |
| MCLR | CN9 – 1 | Microcontroller Master Clear (RESET) pin. |
| Vdd | CN9 – 2 | Positive power supply reference. |
| GND | CN9 – 3 | Negative power supply reference. |
| PGD | CN9 – 4 | Program Data pin. |
| PGC | CN9 – 5 | Program Clock pin. |

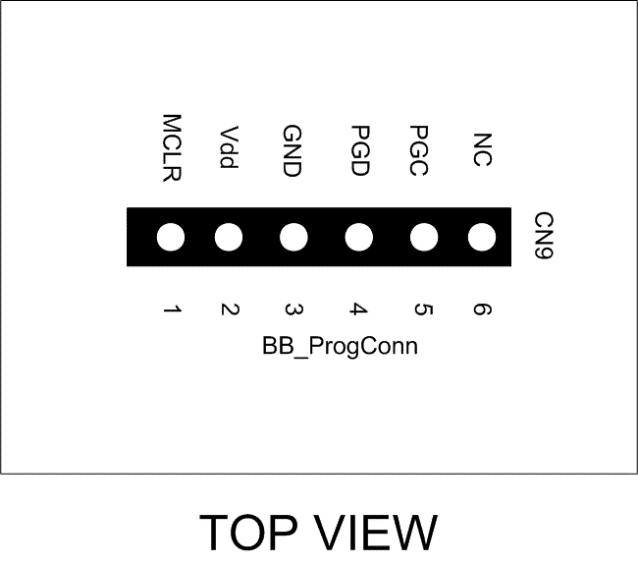


Figure 6 - BB Programmer Connector Pinout

## Communication Connector

The Communcation Connector allow the connection of the communication buses (UART and I2C) to other boards. The connector’s pinout is depicted in Figure 7 and Table 7 explains the meaning of each single pin (NC stands for “Not Connected”).

Table 7 – Comm Connector Pinout

|  |  |  |
| --- | --- | --- |
| Pin Name | Pin Number | Description |
| VddMcu | CN10 – 1 | This pin is connected to MCU regulated positive voltage reference (3,3V). |
| GND | CN10 – 2 | Negative power supply reference. |
| McuTx | CN10 – 3 | BB UART Tx Line. |
| McuRx | CN10 – 4 | BB UART Rx Line. |
| McuSDA | CN10 – 5 | BB I2C SDA Line. |
| McuSDL | CN10 – 6 | BB I2C SCL Line. |

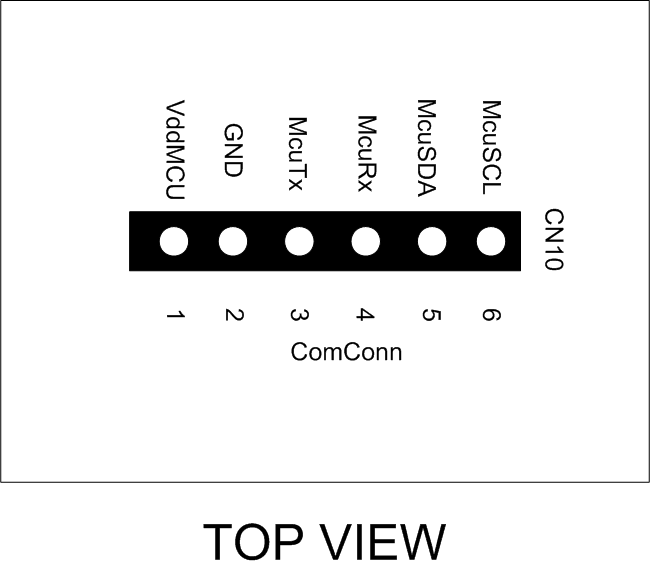


Figure 7 - Comm Connector Pinout

## Expansion Connector

The Expansion Connector allow to connect the EB210 with another Expansion Board. The connector’s pinout is depicted in Figure 8 and Table 8 explains the meaning of each single pin (NC stands for “Not Connected”).

Table 8 – Expansion Connector Pinout

|  |  |  |
| --- | --- | --- |
| Pin Name | Pin Number | Description |
| VddMcu | CN11 – 1 | This pin is connected to MCU regulated positive voltage reference (3,3V). |
| GND | CN11 – 2 | Negative power supply reference. |
| VddBat | CN11 – 3 | This pin is connected to the main power source. |
| INT0 | CN11 – 4 | BB INT0 Line. |
| McuSDA | CN11 – 5 | BB I2C SDA Line. |
| McuSDL | CN11 – 6 | BB I2C SCL Line. |

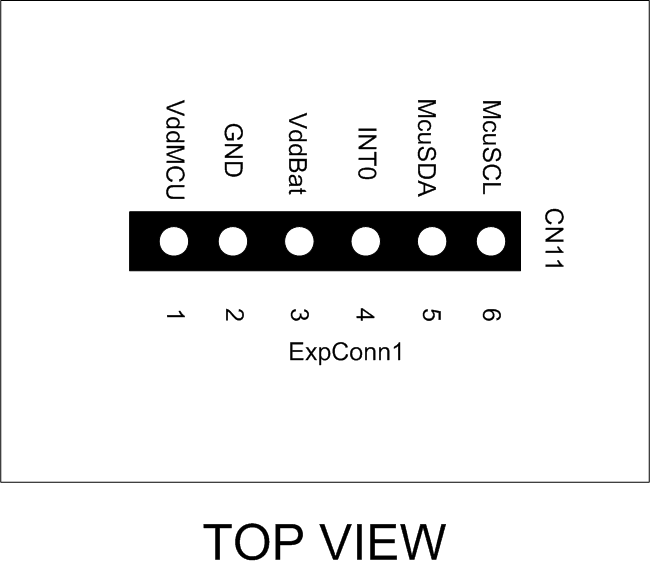


Figure 8 - Expansion Connector Pinout

# Command Set

## Specific Command Set

The EB210 board supports both the MS Generic Command Set (see document MS\_Generic\_Command\_Set) and a set of specific commands (also called Specific Command Set).

Table 9 lists the EB210 Specific Command Set:

Table 9 - Command Set

|  |  |  |  |
| --- | --- | --- | --- |
| Code | Cmd Name | Parameters | Description |
| 0x50 | Display Send Cmd | Cmd (1byte) | Send a command to the LCD display (support the standard HD44780 command set). |
| 0x51 | Display write string | Len (1 byte)  String (16 byte) | Write a string to the LCD display (the first byte must be string length in characters, and the remaining bytes the string to write). |
| 0x52 | Display BL control | BL (1 byte) | Control display backlight (0 OFF, 1 ON). |
| 0x53 | Display Clear Scr | None | Clear the Display. |
| 0x54 | Display Line 1 | None | Move cursor to display Line1. |
| 0x55 | Display Line 2 | None | Move cursor to display Line2. |

## Examples

Some examples of Specific Command Set usage are listed below:

1. Write “Hello World!” to the display: **[0x51] [0x0C] [0x48] [0x65] [0x6C] [0x6C] [0x6F] [0x20] [0x57] [0x6F] [0x72] [0x6C] [0x64] [0x21]**
2. Set Backlight On: **[0x52] [0x01]**
3. Clear the display: **[0x53]**

# Technical Specifications

Table 10 resumes the board technical specifications:

Table 10 - Board Technical Specifications

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Parameter | Max | Typ | Min | Unit | Notes |
| Supply Voltage | 3.6 | 3.3 | 2.0 | V |  |
| Current Cons. (Normal) |  | 10 |  | uA |  |
| Current Cons. (Peak) |  | 1 |  | mA |  |
| Current Cons. (Low Power) |  | 100 |  | nA |  |
| Startup Time |  | 100 |  | mS |  |