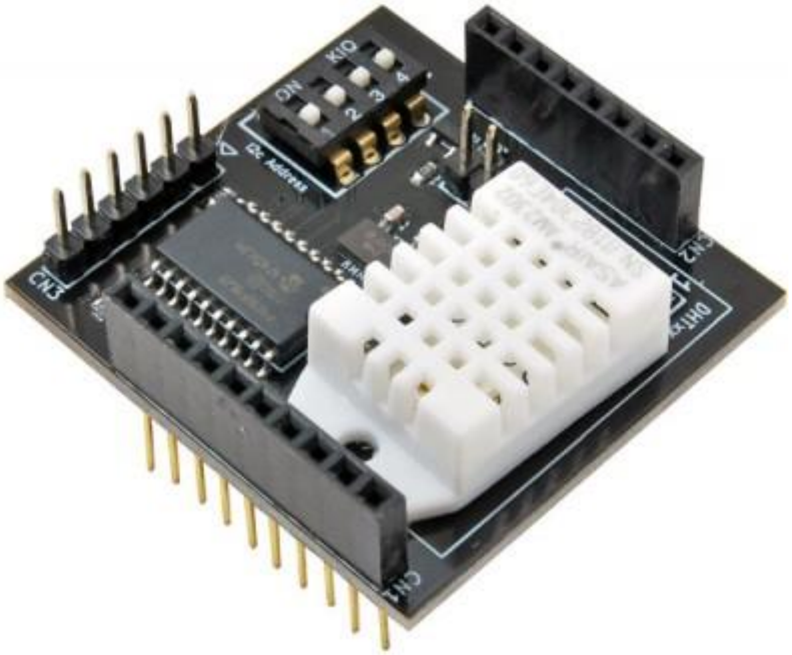
Mercury System

SB330



Temperature Board - Product Datasheet

Author	Francesco Ficili
Date	22/05/2018
Status	Released

Revision History			
Version	Date	Author	Changes
1.0	22/05/2018	Francesco Ficili	Initial Release.
1.1	12/01/2019	Francesco Ficili	Updated images.

SUMMARY

1. 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:

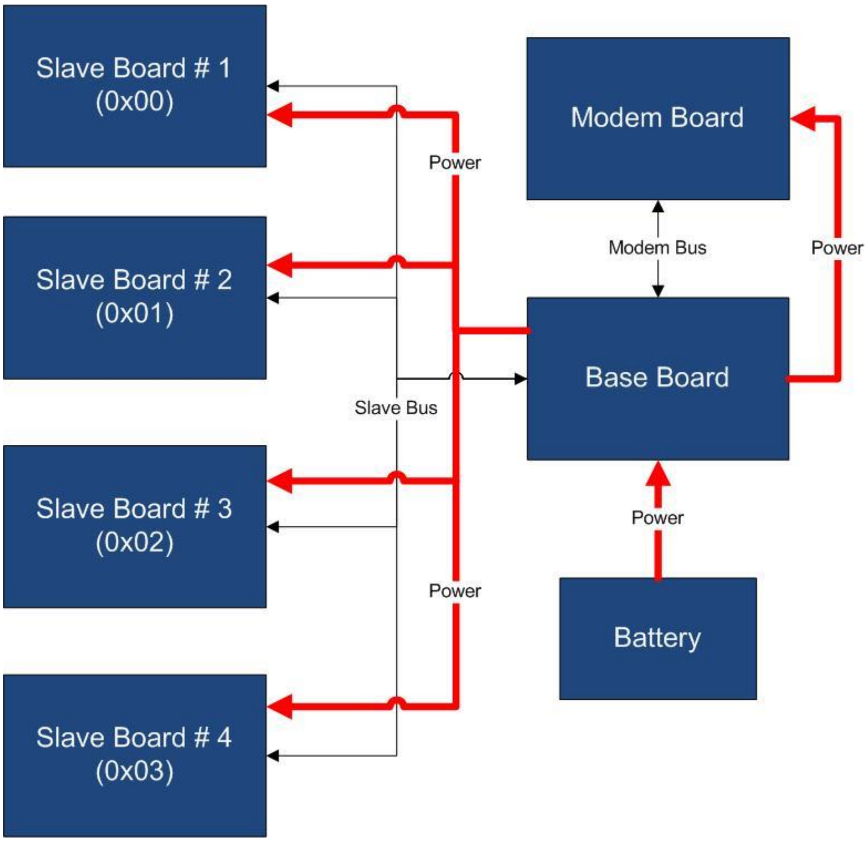


Figure 1 - Typical System Connection

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…

2. Block Diagram

The SB330 is temperature and humidity board, able to interface a DHT22 sensor. Figure 2 shows the SB330 block diagram. The heart of the system is a PIC16F1829 8-bit RISC microcontroller, produced by Microchip Technology Inc.

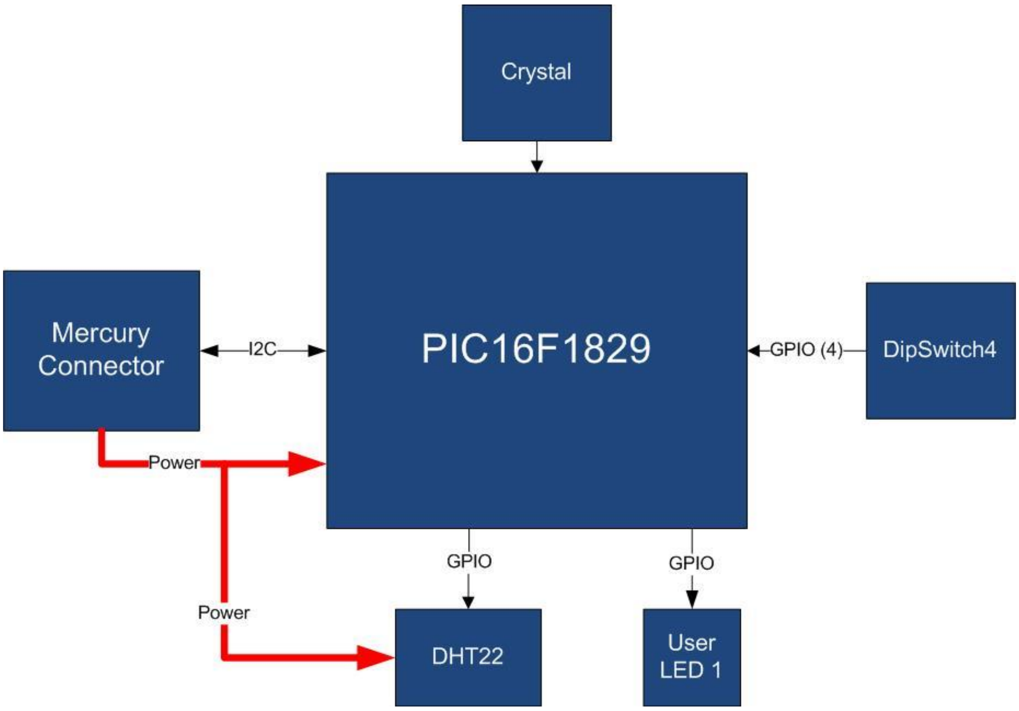


Figure 2 - Block Diagram

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	14
XLP	Yes

The SB330 is connected to a Mercury System Base Board 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). The temperature sensor can measure temperature in the range -40-80 °C, with ± 0,5 °C accuracy and humidity in the range 0-100%, with 2.5% accuracy.

Table 2 resumes the SB330 board main characteristics:

Table 2 – Board Characteristics

Parameter	Description	Notes
Board Type	Slave Board (SB)	
Supported Bus	I2C	
Addressing	Dip Switch 4	
Peripheral Description	Temperature and Humidity Sensor	DHT22

3. Hardware

This section goes deeper in the HW details of SB330. Figure 3 depicts the most important components of the board:

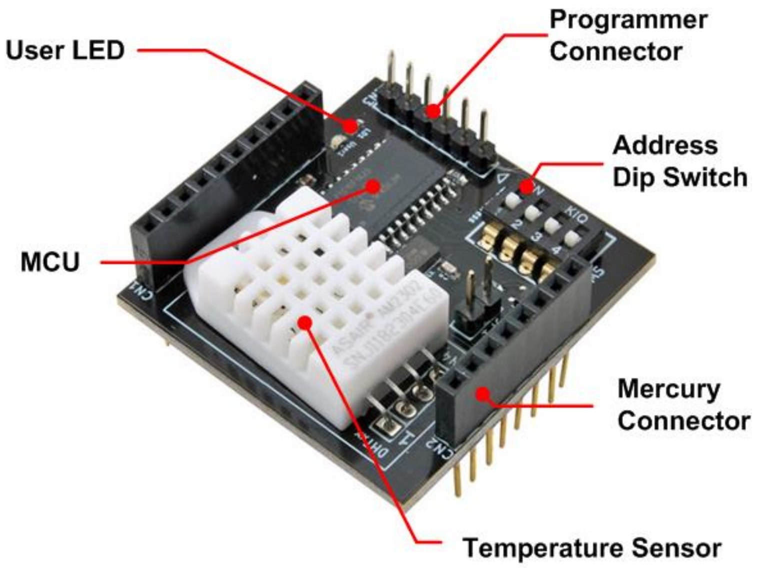


Figure 3 – Hardware Highlight

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

Table 3 – Hardware characteristics

Parameter Name	Description
User LED	Board User LED, by default it's configured as heartbeat LED (periodic pulses).
Temperature Sensor	DHT22 Temperature&Humidity Sensor.
Mercury Connector	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.
Programmer Connector	PicKit 3 Microchip Programmer/debugger connector. It is directly connected to the MCU debug port, in order to allow advanced debugging and programming features, if needed.

4. Pinouts

This section highlights the pinouts of SB330 connectors.

Mercury Connector

The Mercury Connector is the connector which interfaces the SB330 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 – 1 CN2 – 2	This pin is connected to the main power source.
VddMcu	CN1 – 2	This pin is connected to MCU regulated positive voltage reference (3,3V).
GND	CN1 – 3 CN2 – 1	This pin is connected to the board reference voltage.
INT0	CN2 – 3	Interrupt INT0 pin.
SDA	CN2 – 7	This pin is connected to I2C SDA line (Data Line).
SCL	CN2 – 8	This pin is connected to I2C SCL line (Clock Line).

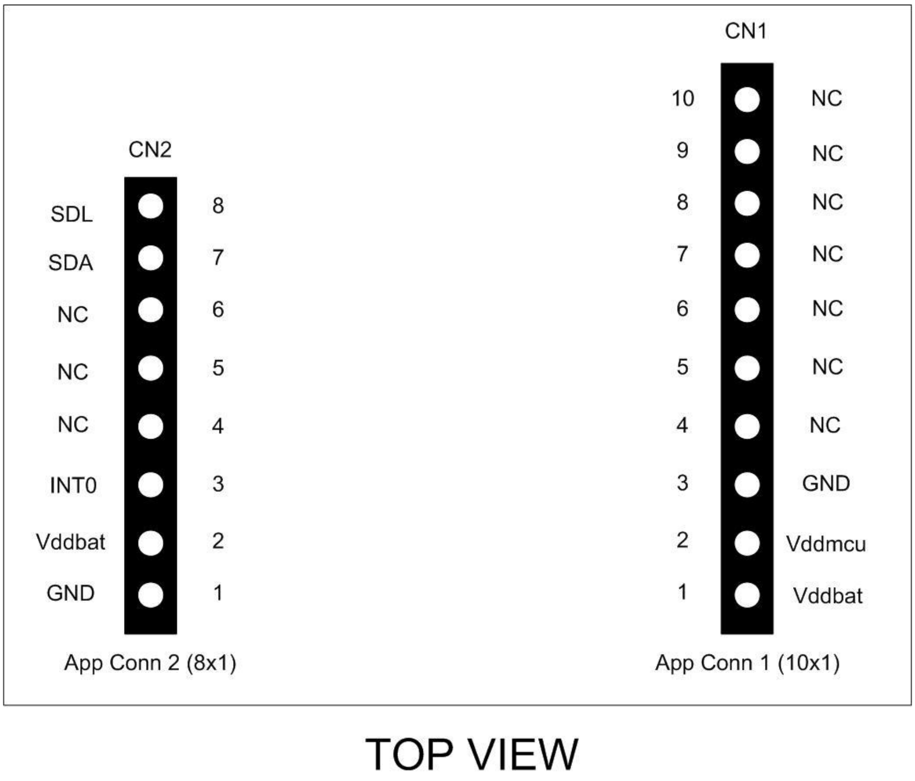


Figure 4 - Mercury Connector Pinout

Programmer Connector

The Programmer Connector is the connector which allows to re-program the SB330 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 - 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	Power ground.
PGD	CN3 – 4	Program Data pin.
PGC	CN3 – 5	Program Clock pin.

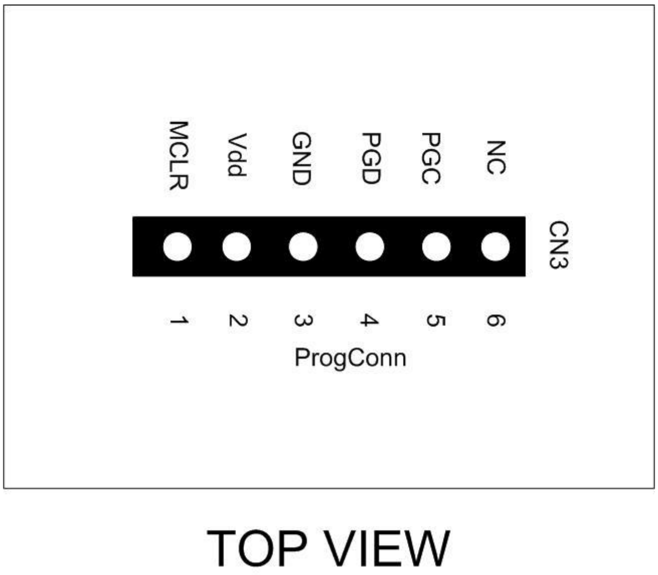


Figure 5 - Programmer Connector Pinout

5. Command Set

Specific Command Set

The SB330 board supports both the MS Generic Command Set (see document MS_GenericCommandSet) and a set of specific commands (also called Specific Command Set).

Table 6 lists the SB330 Specific Command Set:

Table 6 - Command Set

Code	Cmd Name	Parameters	Description
0x60	Request Temperature Raw	None	This command requests the preparation of the temperature data in binary format. The board will prepare 1byte of data containing the decimal part and 1byte of data containing the fractional part of the read temperature value (e.g. 22.5°C will be sent as byte 1 = 22 and byte 2 = 5).
0x61	Request Temperature ASCII	None	This command requests the preparation of the temperature data in ASCII format. The board will prepare 5byte of data containing the temperature value in ASCII format (e.g. 22.5°C will be sent as “22.50”)
0x62	Request Humidity Raw	None	This command requests the preparation of the humidity data in binary format. The board will prepare 1byte of data containing the decimal part and 1byte of data containing the fractional part of the read humidity value (e.g. 60.7% will be sent as byte 1 = 60 and byte 2 = 7).
0x63	Request Humidity ASCII	None	This command requests the preparation of the humidity data in ASCII format. The board will prepare 4byte of data containing the humidity value in ASCII format (e.g. 60,7% will be sent as “60.7”)
0x64	Request cks status	None	This command requests the preparation of the checksum data in binary format. The checksum is calculated over the sensor data at each data refresh (which happens each 2s). If the checksum is ok, then the data is updated, otherwise the previous values are kept. This value can be used to understand if the last value read is fresh or not.

Examples

Some examples of Specific Command Set usage are listed below:

1) Request temperature data in binary: **[0x60] + 2byte Read Request**
2) Request humidity data in ASCII: **[0x62] + 4byte Read Request**

6. Technical Specifications

Table 7 resumes the board technical specifications:

Table 7 - Board Technical Specifications

Parameter	Max	Typ	Min	Unit	Notes
Supply Voltage	3.6	3.3	2.0	V	
Current Cons. (Normal)		2		mA	
Current Cons. (Peak)		3.5		mA	
Current Cons. (Low Power)		100		uA	
Startup Time		100		mS	
Temperature Range	80		-40	°C	
Temperature Accuracy		±0.5		°C	
Humidity Range	100		0	%	
Humidity Accuracy	5		2	%	