

# User Manual for Aurum v1.2

Deepak Bhat  
Ashwin Radhakrishnan  
Debapratim Ghosh

Microcontroller-based Development Team  
Virtual Labs (e-Prayog) Wadhvani Electronics Lab  
Indian Institute of Technology Bombay

April-May 2012

## 1 Board Synopsis

Aurum v1.2 (hereafter referred to as Aurum) is a PIC18F4550-based board developed at e-Prayog, WEL Lab, IIT Bombay under the Virtual Labs Project by the Ministry of Human Resource and Development (MHRD), Government of India. This is a low-cost solution aimed to cater to the needs of an undergraduate electrical/electronics engineering student in a first course in microprocessors, microcontrollers and embedded systems.

## 2 Prerequisites for using Aurum

Before using Aurum, you are expected to have a basic familiarity/ working knowledge of:

- Microprocessors (8085 family), microcontrollers (8051 family)- programming model, I/O handling and other general functions.
- Familiarity with programming in C.

Some extra knowledge which would help you (which can be learned while using Aurum as well):

- Familiarity with devices in the PIC18 family.
- Brief idea of various interfaces- parallel and serial I/O, I2C, SPI etc. and other peripheral interfaces used in embedded systems.
- Brief idea of on-chip peripherals available in a general PIC18 MCU.

An overview to the PIC18F4550 is provided on the e-Prayog webpage, and a comprehensive datasheet is provided by Microchip Technology (the manufacturers of the PIC family).

### 3 Contents in the Support Disk

The support CD along with Aurum has the following contents.

- Aurum user manual (obviously!)
- Example programs with manuals (labsheets)
- Installer executables for MPLAB v8.84 and Microchip C18 Toolsuite v3.42
- HID Bootloader v2.6a
- Video Tutorials about the board, and using MPLAB and HID Bootloader

### 4 Precautions for using Aurum

- Do not touch the pins of any onboard IC directly, to avert the risk of damage by electrostatic discharge.
- When interfacing with large number of external peripherals, power the board through a single +5V DC supply (tested).

### 5 Features in Aurum

Figure 1 shows a top view of Aurum, with the various on-board resources labeled.

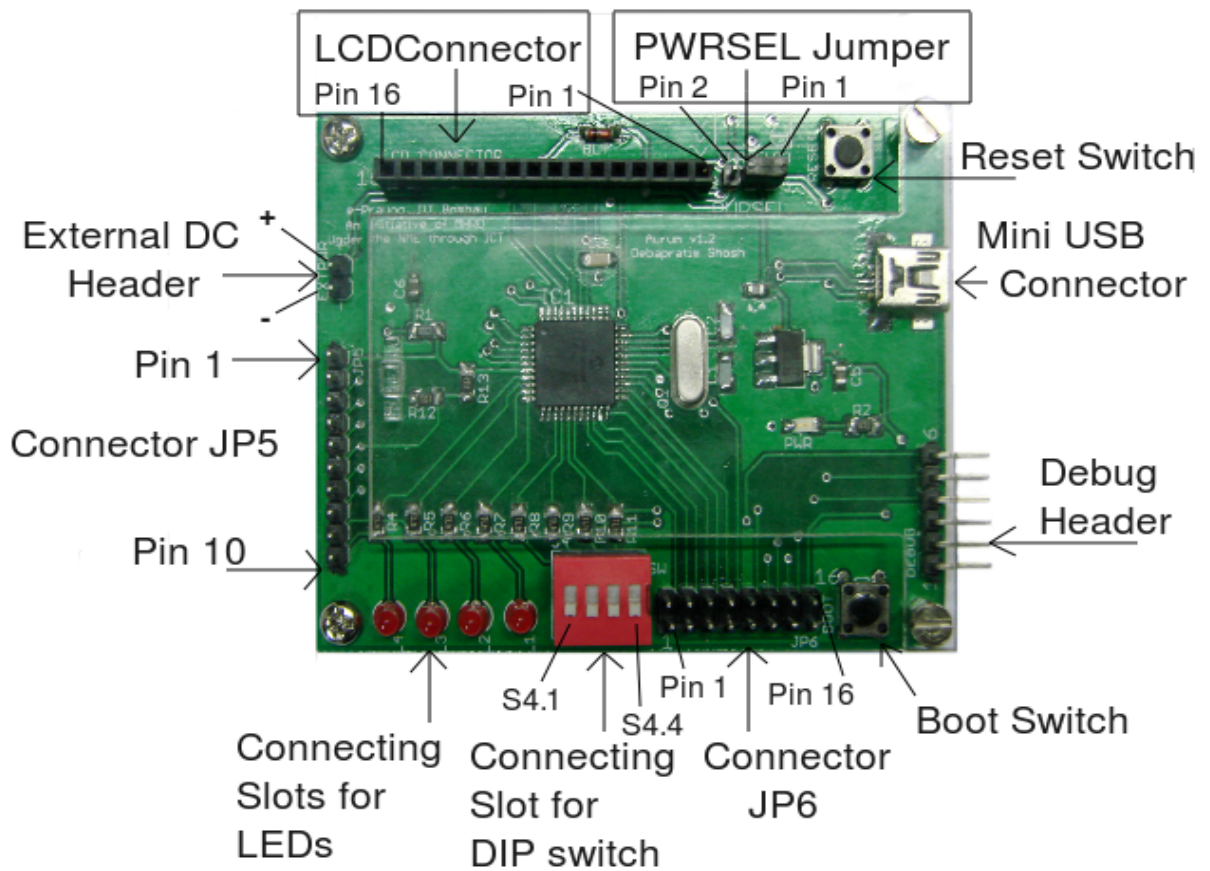
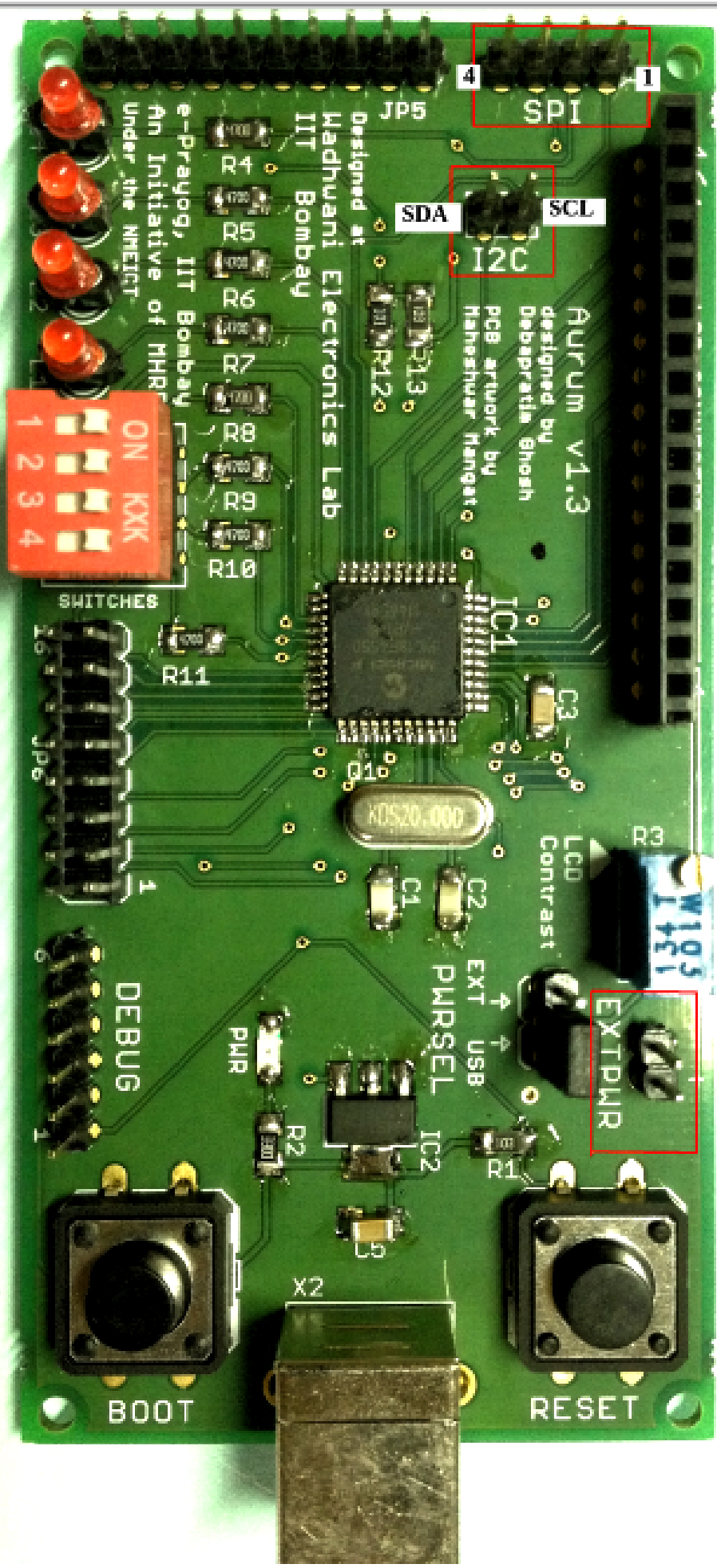


Figure 1: Aurum top-view with labeled features



### SPI- Connections

- 1 - SCK
- 2 - SDI
- 3 - SDO
- 4 -  $\overline{SS}$

# Aurum v1.3

Following are the features on Aurum.

- A board centered around PIC18F4550, an 8bit microcontroller by Microchip Technology.
- USB powered and programmable. Additionally, a provision to power through external power supply is also provided (*See* Section 3).
- Large number of on-board I/Os (31) are provided for various applications.
- Connectors are made available for the interfacing of standard I/O such as switches/LEDs and LCD.
- Compatible with standard Microchip Technology software tools (MPLAB IDE and HID Bootloader).

## 6 Configuring Aurum Before Programming

You will need to ensure that an appropriate power source is selected for the MCU while programming/ running an application. To do this, an on-board jumper PWRSEL needs to be connected suitably.

PWRSEL Jumper Setting	Power Source For Aurum
1. Between pin 1 and center	Board powered via USB (Bus-powered via host PC)
2. Between pin 2 and center	Board powered through external power supply header <b>EXTPWR</b> .

Table 1: Configuration for PWRSEL jumper

## 7 Getting Aurum working with your PC

1. Plug in the USB cable into the mini-USB connector on Aurum, and the other end to a USB port on your PC. The power LED should glow.
2. The PIC18 MCU on Aurum contains an on-chip bootloader (programmed during board fabrication) which communicates with the host PC. This is an HID bootloader, hence it does not need any driver installation, and automatically gets appended to the Human Interface Devices (HID) list of devices connected to the PC.
3. Navigate to the HID-Bootloader folder in your PC and open the HIDBootloader.exe executable. A window indicating loader functions should open up.
4. If the MCU is in the bootloader mode, then the status area of the loader should show a message **Device Attached**. Also, the loader function tabs such as Program/ Verify/ Reset Device etc. should be active.
5. If the status on the loader shows **Device removed** or **Device not recognized**. **Verify device is in bootloader mode**, hit the **BOOT** switch, and then simultaneously press **RESET**. Hold for about 1-2 seconds. Then, release **RESET** and release **BOOT**. Follow the order press **BOOT** → simultaneously press **RESET** → release **RESET** → release **BOOT**.
6. Once the status reads **Device attached.**, the device is in the bootloader mode. The board is now ready to be programmed.

## 8 Using MPLAB IDE to make Applications

MPLAB is the IDE provided by Microchip Technology for the user to write their assembly/C programs for a PIC MCU target to make an application. MPLAB has a free downloadable version that can at least, meet the requirements of an undergraduate curriculum. This document primarily focuses on C programming. Make sure that along with MPLAB, the Microchip C18 compiler is also installed. Follow the steps given below to make and application.

1. Open MPLAB IDE. A blank window should open up. Go to Project → Project Wizard. Select the target device as P1C18F4550.
2. Select the active toolsuite as Microchip C18 Toolsuite. This is the toolchain i.e. the compiler, linker etc. which you will use for this project.
3. In a separate directory, open a project file (of the extension `.mcp`).
4. You need not add existing files to your project (especially when you are creating a new project). Skip the add files step- you can always add existing files at a later stage.
5. Finally, before creation, the project summary will be shown. Verify the details before you click on Finish.
6. Now, you can write your C code. Go to File → New. A text editor opens up. Save it as `filename.c`.
7. Write your program in this editor. Do not forget to include the interrupt vector relocations at the start of the code (see the example programs in the e-Prayog webpage/ support disk).
8. Once your code is complete, navigate to the project workspace in MPLAB (`projectname.mcw`), right click on Source Files, then click on Add File, and add your program file (C file) to the workspace. More source files can be added likewise (for larger projects).
9. Next, right click on Linker Script (in the same workspace) click on Add File, and add the linker script file `rm18f4550.lkr` to the workspace (provided in the support disk/webpage).
10. Now go to Project → Build All. This will compile and link the files in your project and build an executable (`.hex`) file. It will be saved as `projectname.hex` in the same project directory.

## 9 Loading the Application into Aurum

To load the application executable (the HEX file created), the MCU must be in the bootloader mode. You can test the same by following the instructions in Section 6. To transfer the HEX file to the MCU, follow the given steps.

1. Open the `HIDBootloader.exe` application in the `HIDBootloader` folder in your PC (assuming the MCU is in the bootloader mode).
2. Click on `Open Hex File`, browse to the directory containing your HEX file, and double-click it.
3. Click on `Program/Verify`. This loads the application HEX file into the MCU's program memory.
4. Now, reset the device to bring it in the application mode. This can be a hardwired reset (by pressing the `RESET` switch) or a software reset (by clicking `Reset Device` in the `HIDBootloader.exe`).

5. You should now be able to see your application running.

## 10 Using Aurum for Further Applications

Connectors are provided on Aurum to use for commonly applied peripherals (such as LEDs, switches, LCD etc.) as well as for general-purpose I/O. This section describes how you can use these connectors for various applications.

1. The LCD Connector: This is a 16pin female vertical connector on the board, marked `LCD_CONNECTOR`. You can interface a standard  $16 \times 2$  character LCD module (most commonly, the JHD162A). The pin 1 of the LCD module should be connected to the pin marked `>>>` on the board. On `R3`, insert a  $10k\Omega$  pot for adjusting the display contrast.
2. The switch/LED connectors: Provisions are kept for connecting switches and LEDs. The connector `S4` can be used to interface standard DIL/ Topaz switches. The connectors marked `L1 . . . L4` can be used to interface 3 mm LEDs. The LEDs are to be placed with their cathodes facing the board edge.
3. While interfacing a large number of I/Os, it is recommended to use an external power supply as the USB port can source only about 400 mA of current. In such a case, change the `PWRSEL` settings appropriately as mentioned in Section 5.
4. Most importantly, your application software is dependent on the MCU's I/O pinouts. Shown below are the connector pinouts as on the board.

LCD_CONNECTOR pin no.	Pin of PIC18F4550 connected to
1 (>>>)	GND
2	+5V DC (Note 1)
3	-
4	RE0/AN5/CK1SPP
5	RE1/AN6/CK2SPP
6	RE2/AN7/OESPP
7	RD0/SPP0
8	RD1/SPP1
9	RD2/SPP2
10	RD3/SPP3
11	RD4/SPP4
12	RD5/SPP5/P1B
13	RD6/SPP6/P1C
14	RD7/SPP7/P1D
15	-(Note 2)
16	-

Table 2: Pin-outs for `LCD_CONNECTOR`

Connector JP5 pin no.	Pin of PIC18F4550 connected to
1	GND
2	VDD (+3.3V)
3	RB7/BKI3/PGD (Note 3)
4	RB6/KBI2/PGC
5	RB5/KBI1/PGM
6	RB4/AN11/KBI0/CSSPP
7	RB3/AN9/CCP2/VPO (Note 4)
8	RB2/AN8/INT2/VMO
9	RB1/AN10/INT1/SCK/SCL
10	RB0/AN12/INT0/FLT0/SDI/SDA

Table 3: Pin-outs for connector JP5

Connector JP6 pin no.	Pin of PIC18F4550 connected to
1	VDD (+3.3V)
2	GND
3	RA0/AN0
4	RA1/AN1
5	RA2/AN2/VREF-/CVREF
6	RA3/AN3/VREF+
7	RA4/T0CKI/CIOUT/RCV
8	RA5/AN4/SS/HLVDIN/C2OUT
9	RA6/OSC2/CLKO
10	GND
11	RC2/CCP1/P1A
12	RC7/RX/DT/SDO
13	RC1/T1OSI/ICCP2/UOE
14	RC6/TX/CK
15	RC0/TIOSO/T13CKI
16	VDD (+3.3V)

Table 4: Pin-outs for connector JP6

Note:

1. +5V DC Line is used as power supply for the LCD. While using an external power supply to drive LCD, take care to give no more than +5V on this pin.
2. Pins 15 and 16 on the LCD\_CONNECTOR are used for backlight supply.
3. Pins 3-6 on connector JP5 are connected to the S4 switch connector in this order: JP5.3 → S4.4, JP5.4 → S4.3, JP5.5 → S4.2 and JP5.6 → S4.1.
4. Pins 7-10 on connector JP5 are connected to the LED slots in this order: JP5.7 → L1, JP5.8 → L2, JP5.9 → L3 and JP5.10 → L4.

Figure 2 shows the pin numbering convention for connector JP6.

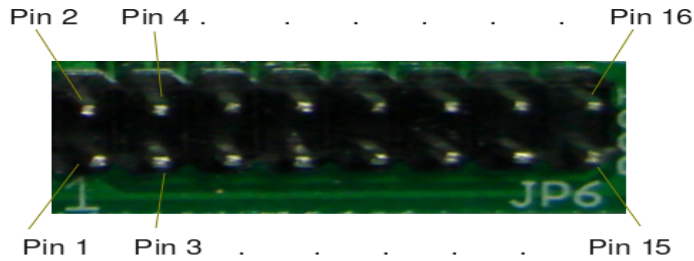


Figure 2: Pin numbering convention for connector JP6

## 11 Frequently Asked Questions (FAQs)

1. From where can I download MPLAB IDE and which version should I use?  
 Ans: You can visit the following URL:  
[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en023073](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en023073)  
 The latest version as at the time of writing this manual is MPLAB v8.84. Alternatively, you may use MPLAB X.
2. While creating a new project in MPLAB, the Microchip C18 Toolsuite is not listed/ shows crosses 'x'. What should I do?  
 Ans: Mot likely, the toolsuite has not been installed. You can visit the following URL:  
[http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=1406&dDocName=en010014&part=sw006011](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=1406&dDocName=en010014&part=sw006011)  
 It contains the latest version, which you can download and install. In case the C18 tool-suite is installed and the problem is persisting, make sure you have chosen the correct target device for the project.
3. Can I debug my program?  
 Ans: Yes. You can run a software simulation using MPLAB SIM. Hardware emulation may also be done using a PIC emulator (Pickit 3, for instance). Contact us for help.
4. The `HIDBootloader.exe` shows an error while opening. What could be the problem?  
 Ans: This executable requires Microsoft .NET Framework v2.0 (or higher) installed on your PC to run. If you do not have this distribution, you can download it for free from Microsoft.
5. What system requirements do I need to get my PC working with Aurum?  
 Ans: You need a minimum of Windows XP (Service Pack 2 or higher) with the Microsoft .NET Framework v2.0 installed on your PC.
6. Can Aurum function as a standalone application unit?  
 Ans: Yes. After the MCU is successfully programmed, you can deploy it in any application. In such cases, the power should be given through a minimum of +5V DC supply (a battery, for instance). Make sure you have the MCU running in the application mode while deploying.
7. Is USB the only method to program the device?  
 Ans: No. You may use a PIC emulator (purchased from Microchip) to program the MCU. The `DEBUG` header is provided for that purpose. Of course, USB makes it much faster and economical to program!

8. Are the LCD/Switch/LED connectors dedicated?

Ans: No. The switch/LED connectors are multiplexed as explained in Section 9. The LCD connector can be used as general-purpose I/O in case the application does not require the use of an LCD. Note that while connecting the above devices, no extra circuitry need be designed for their interfacing.

9. How do I use a PIC emulator with Aurum?

Ans: Simple. You can program/ debug using the emulator. However, this will erase the USB bootloader already present on the MCU. To make it usable for USB programmability, you will need to reprogram the bootloader on to the MCU.

10. How do I handle the configuration bits?

Ans: The bootloader has certain configuration bits set, which will apply to the application program by default. This is the case when in MPLAB, you go to Configure → Configuration Bits and click on the checkbox **Configuration Bits set in Code**, but do not explicitly define those in your application code. If you explicitly define configuration bits in your program, those settings will take effect at run-time. However, this does not affect the bootloader configuration settings in any way.

Alternatively, if you uncheck the **Configuration Bits set in Code** checkbox, the configuration settings in the Configuration Bits window will take effect.

## 12 Help and Support

You are welcome to contact the Microcontroller-based Development Team at e-Prayog IIT Bombay to report software bugs, hardware failures/ other issues, and help and information in application developments. Suggestions regarding mistakes in the manual and other documentation/ distributables are welcome. Various queries and hate mail are also welcome.

Send an e-mail to: [uc.vlabs.wel@gmail.com](mailto:uc.vlabs.wel@gmail.com)

For all administrative queries, send an e-mail to: [vlabs\\_wel@ee.iitb.ac.in](mailto:vlabs_wel@ee.iitb.ac.in)

Some helpful links are as follows.

1. <http://59.181.142.81/>. This is the link to the e-Prayog webpage. Create a free login for yourself to have access to all our developments.
2. [ww1.microchip.com/downloads/en/devicedoc/39632c.pdf](http://ww1.microchip.com/downloads/en/devicedoc/39632c.pdf). This is the link for the PIC18F4550 datasheet, which you will need while working to develop your own projects.
3. [http://www.microchip.com/stellent/idcplg?IdcService=SS\\_GET\\_PAGE&nodeId=2680&dDocName=en537044](http://www.microchip.com/stellent/idcplg?IdcService=SS_GET_PAGE&nodeId=2680&dDocName=en537044). This is the link for Microchip Application Libraries, a wide variety of applications, including the USB HID Bootloader PC software.