PCB_111000_UNO part 2: Assembly and programming the UNO and PCB_A

Part 1 Building PCB111000_1
Details are given in PDF file "Parts list_and_assembly" contained in the attachment pcb_design_and_assembly.
Note:

The location of the PCB111000_1 reset switch.
User switches sw1, sw2 and sw3.
The links that determine whether or not common anode or common cathode displays are to be used.

It is often required to double click the reset switch. Where this is the case the double click should take place within 300mS.

Unlike the UNO, the PCB111000_1 device is clocked from its internal oscillator which may need calibration to ensure reliable communication with a PC. This is done automatically during the system programming stage and should not need to be done again.

Part 2 Unzip the system programs
Zip file "WinAVR_and_UNO_files" contains the following files

I2C_V18_CA_part_1.hex
I2C_V18_CA_part_2.hex
Prog_plus_Verification.hex
Found using the path "1_PCB_A_Mini_OS_I2C_V18/4_hex_files/CA_Version/"
Project_programmer_UNO
Hello_world.txt
One UNO project and one AVR project.

Hex files are also present for use with CC displays
The "UNO_bootloader_for_hex&text_V6" hex file saved in part 1 of the project will also be required.

Part 3 Uploading "Project_Programmer_UNO"
Connect PCB 111000_UNO to the USB port of a PC.
Open Project_Programmer_UNO and upload it.
Open a terminal program (Bray@++ is recommended),
Set it to 57600 Baud rate, 1 stop bit, 8 data bits and no parity or handshaking.
User prompt "s……." should be displayed on the screen.

Part 4 Programming the PCB111000_1 device with the mini-OS
Press "s", then "p" at the user prompt and upload the three hex files
Start with "part_1" and send "part_2" as soon as the LED stops flashing
End with " Prog_plus_Verification"
Press "0" at the "Integer(0-FF)?" prompt.
Note that the PCB111000_1 (also known as PCB_A) device is automatically calibrated at this time.

When the prompt "s……." returns press "s", "e" and "w".
Press "0" at the "Integer(0-FF)?" prompt.
Send the "Hello_world" text file four times and press any key at the "AK?" prompt.
The file will be echoed to the screen.

Part 5 Uploading the UNO project
Close the terminal program
Open “Proj_2C1_random_LEDs_UNO /Proj_2C1” and upload it.
The display is driven by a random number generator.
This tests the operation of the display
It also removes “Project_Programmer_UNO” from the UNO device

Part 6 Reprogramming the UNO device
Re open the terminal program.
Double click the PCB_A reset switch.
Press “p” and the “?” prompt should be displayed.
Press “r”and the Arduino code will be downloaded to the screen.
It should be saved so that it can be restored at a later time if required.
Double click PCB_A reset switch again.
This time press “p” twice.
At the “UNO_update?” prompt send
“2_UNO_bootloader_for_hex&text_V6/Hex_files/Full_Hex_CC_and_CA.hex”
Press “0” at the “Integer(0-FF)?” prompt.
User prompt “h/t/r/D……..” will be displayed on the screen.

Part 7 Testing the default programs
At the prompt h/t/r/D press “r”.
The display should be driven by a random number generator
Hold sw1 down and pulse sw2.
Strings from the “Hello world” file should be printed out one at a time.
Hold sw1 and sw3 down together and it should be possible to adjust the display brightness.
Note:
A single pulse of PCB_A reset switch should momentarily pause the display.
A double click should halt it and restore the “h/t/r/D……..” user prompt.
Power cycling should also restore the program

Part 8 Uploading the sample user program and commentary
After power cycling take care to double click on the “Disconnect” button on the terminal program.
Double click the PCB_A reset to restore the “h/t/r/D……..” user prompt.
Press “t”.
Send the commentary file and immediately afterwards send the hex file.
At the “0 to verify or AOK” prompt press “0” twice.
Press “r” at the user prompt and the commentary will be printed out one line at a time.
Press “X” and the program will run.

Note: the commentary is only printed out immediately after the program has been uploaded
It it is not wanted at all press “h” at the user prompt and just upload the hex file.

Key press “T” is provided for use when developing commentary files.

Display intensity can be adjusted while the user program is running:
Hold sw1 and sw3 down and pulse pcb_a reset switch.

PCB_A device can also be recalibrated:

Hold sw1 and sw2 down, pulse pcb_a reset switch and send “r” at the user prompt.

At the “h/t/r/D……..” user prompt pressing “D” will restore the default program.

Power cycle PCB111000_UNO and the user/default program will run immediately.

Part 9 Restoring the Arduino SW

Double click the pcb_a reset switch and press “p” followed by “q”.
At the “Restore Arduino code?” prompt upload the Arduino bootloader code that was saved earlier (part 6).

Note: The Arduino bootloader code starts at address 0x7E00.
When delivered the UNO device also has a short section of code at address zero.
This cannot be restored using this project however the operation of the bootloader does not appear to be seriously impaired.

Part 10 Potential problems

Uploading the wrong system code can result in the UNO having no bootloader, or pcb_a device having no programmer. Various approaches have been tried to deal with this problem:

Recovery code has been written.
Boards can be swapped between two PCB11100_UNOs.
A DIL socket can be used to connect the Atmega 328 device to PCB_A.
“Project_programmer_UNO” has an option to delete PCB_A device EEPROM

The main reason for the problem in recovering from this situation is that both the UNO and PCB_A require access to the USB bridge supplied by the UNO. To avoid latch-up the Tx USARTs of these devices must never be active at the same time. There is no obvious way to use a DPDT switch to isolate the Tx ports without modifying the UNO in some way.

The main conclusion is:
Take time and care when setting up PCB_111000_UNO and remember that 300mS is not much time in which to double click the PCB_A reset switch.