PCB_111000_V2: Getting a new pcb up and running

1. Introduction

PCB_111000_V2 is a development of PCB_111000.
Both pcbs are loaded with

An Atmega 168 which hosts user projects

An Atmega 328. This hosts a mini-OS which:

- Drives an 8 digit display
- Hosts a pcb bootloader that programs the Atmega 168
- Provides a variety of services for use by user projects

Differences are:

For the HW

- The FETS used to drive the display are no longer fitted

For the SW

- User projects have been revised and are now compiled using Arduino
- System programs are compiled using Studio 7
- PCB initialisation is carried out using the following Arduino sketches rather than bespoke HW
  - 8_UNO_AVR_programmer_V3
  - 5_Project_pcb_168_V2.30_Arduino_V2

2. Setting up PCB-111000_V2

Step 1 The PCB is loaded with all components except the displays and the Atmega devices

Step 2
Open the sketch “8_UNO_AVR_programmer_V3”. Under Tool/Board select Arduino UNO.
Connect the UNO to the PC check the port number and upload the sketch.

Step 3
Close the sketch assemble the HW (see section 3) and with the aid of the Br@y++ terminal emulator use the UNO to:

- Upload “5_Project_pcb_168_V2.30_Arduino_V2” to the Atmega 168
- Calibrate it’s internal RC oscillator
- Allow the Atmega 168 to run so that it can upload string data to its EEPROM.

Note:
Terminal emulator runs at 38.4KB and uses default settings.
See section 4 for more details.

Step 4 Load the Atmega 168 device onto the PCB111000 and test, then add the Atmega 328.

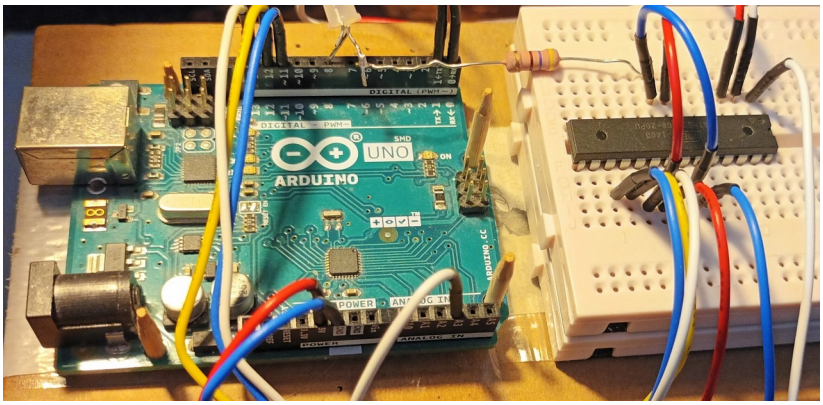
Step 5
Run “5_Project_pcb_168_V2.30_Arduino_V2” under the control of Br@y++ to setup the Atmega 328 (See section 5 for more details) by:

- Uploading the mini-OS to its flash
- Checking the calibration of its internal RC oscillator.
- Uploading the “Hello world” strings to its EEPROM

Step 6 Overwrite “5_Project_pcb_168_V2.30_Arduino_V2” with user projects. (See sections 6 and 7.)

Step 7 Upgrade the mini_OS if necessary (See section 8).

3. Hardware (HW) details



UNO used to program the Atmega168 before it is soldered onto the PCB111000_V2

Note: Atmega328 pin numbers given for a DIP28 device
Remove Atmega168 before uploading “8_UNO_AVR_programmer_V3” sketch

UNO			Plug-in pcb
UNO Pin	Atmega 328		Atmega168
	Pin	Function	Pin
0 (Rx)	2	RXD	2
1 (Tx)	3	TXD	3
11	17	MOSI	17
12	18	MISO	18
13	19	SCK	19
A3	26	PC3	1 (Reset)
GND			8, 22
5V			7,20
8	14	PB0	LED driver

4. Driving “8_UNO_AVR_programmer_V3”

Having uploaded the sketch to a UNO open Br@y++ and set it up as follows:
8 data bits, 1 stop bit, no parity or handshaking and a baud rate of 38.4KB

Click on “scan” to identify the comm port and then click on connect:

The following user prompt should be generated:
“s s s s s s s……”

Step 1 Click mouse pointer in the text box (at the bottom of the Br@y++ window) and press -s- to get
“Atmega 168 detected.
Press -p- to program flash, -r- to run target, -d- to clear the target EEPROM or -x- to escape.”

Step 2 Press -p- and send “5_Project_pcb_168_V2.30_Arduino_V2.ino.eightanaloginputs”

Step 3 Follow instructions to Cal device then get user prompt “P/S P/S P/S……”

Step 4 Press -s-, change bit rate when asked and send Text_files\on-chip_strings.txt

A bug Sometimes random text following programming. Power cycle and reprogram.

Optional steps
At P/S prompt press -v- to get program name plus version “Project_pcb_168_V2.30_Arduino”

Reset UNO and change the BR to 38.4kB to restore user prompt “s s s s s s s…….”
Press -s- then -d- to clear “on-chip_strings” and cal bytes from Atmega168 EEPROM

Reset UNO to restore user prompt “s s s s s s s…….”
Press -s- then -t- to recalibrate the Atmega 328 and restore user prompt “P/S P/S…...…”

Reset UNO to restore user prompt “s s s s s s s…….”
Press -s- then -r- to restore user prompt “P/S P/S P/S…….”

Press -s- and reprogram the on-chip strings

Reset UNO to restore user prompt “s s s s s s s…….”
Press -s- then -x- to escape

Reprogram the Atmega168 and note that it now offers a recalibration

5 Driving “5_Project_pcb_168_V2.30_Arduino_V2”

5.1 Add the Atmega 168
Solder the Atmega 168 to the PCB-111000_V2.
Press the DPDT switch left
Connect the pcb to a PC running Br@y++ running at 57.6KB.
Check the port and the user prompt “P/S P/S P/S……..” should be generated.
Press -p- and the response TTND (Target not detected) should also be generated.
This confirms that the USB port is working correctly.

5.2 Add the Atmega 328
Solder on the Atmega 328 and press -p- again to generate the prompt:
“ATMEGA 328 detected
Press -P- to send a program file -E- to send a text file (or X to escape).
p/e p/e p/e p/e p/e p/e…..”

5.3 Upload the mini-OS to the Atmega 328
Press -p- and send
Mini-OS_V2\Hex_files\CC_files\Mini-OS_V26_CC.hex
When the LED stops flashing send 2_pcb_Bootloader_V4_28_CC.hex

Note: This assumes that Common Cathode displays are to be used. Similar files are present for Common Anode displays.

5.4 Auto cal the Atmega 328
Follow instructions to auto cal the Atmega238 and restore the “P/S……” prompt.

5.5 Upload the “Hello_world” file to the Atmega 328 EEPROM
Press -p- then -e- then -w- and send “Text_files\Hello_world.txt several times as requested.
Finally press any key to read back the file from EEPROM.

6. Run simple User Project

Press the DPDT switch right to get the "p/r p/r p/r ….." (May need to press the pcb reset switch.)
Press -p- and send
"9_Nano_Projects\System_programs\Proj_9F_Test_Proj_1\
Proj_9F_Test_Proj_1.ino.eightanaloginputs"

At the "p/r p/r p/r ….." prompt press -r- and operate the DPDT switch.
A column of numbers will be printed out.
Press sw1 to repeat.

Operate the DPDT switch and press the reset switch to restore the "p/r p/r p/r ….." prompt.
This confirms that the Atmega328 is being programmed properly and the displays can be added.

7. Check the displays

Displays are driven using a multiplexer which can make fault finding difficult.
Repeat as for section 6 but send "Proj_9G_Test_PCB_Assembly.ino.eightanaloginputs". This turns off the multiplexer and illuminates digits one at a time.
Check for dry joints or solder bridges if any segments fail to work.

8. To upgrade the mini_OS

Push the DPDT switch right and press the reset switch.
Press -p- at the "p/r p/r p/r p/r p/r…." user prompt
and send 5_Project_pcb_168_V2.30_Arduino_V2.ino.eightanaloginputs
Run the project to get the "P/S P/S P/S……."
The new mini-OS can now be uploaded as described in section 5.3

9. PCB assembly

Eagle circuit design and layout drawings can be found under PCB-111000_V2\PCB 111000_Eagle_drawings together with a parts list.

Photographs of the pcb are shown below.

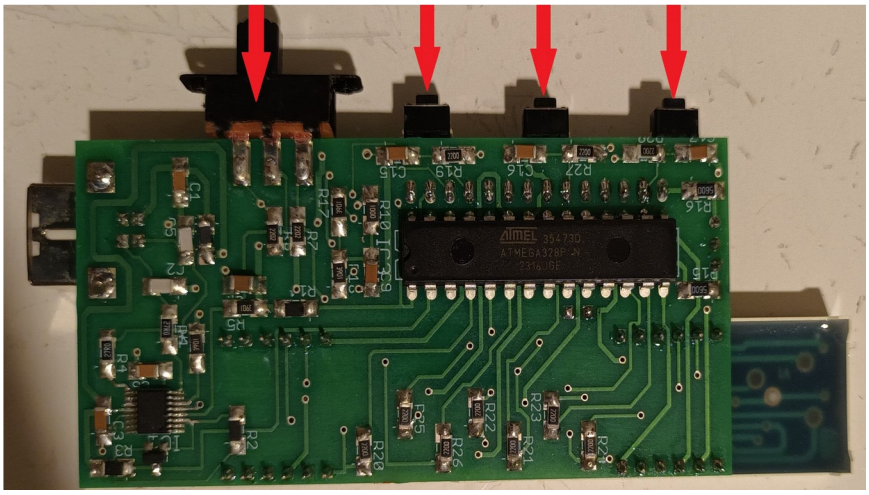
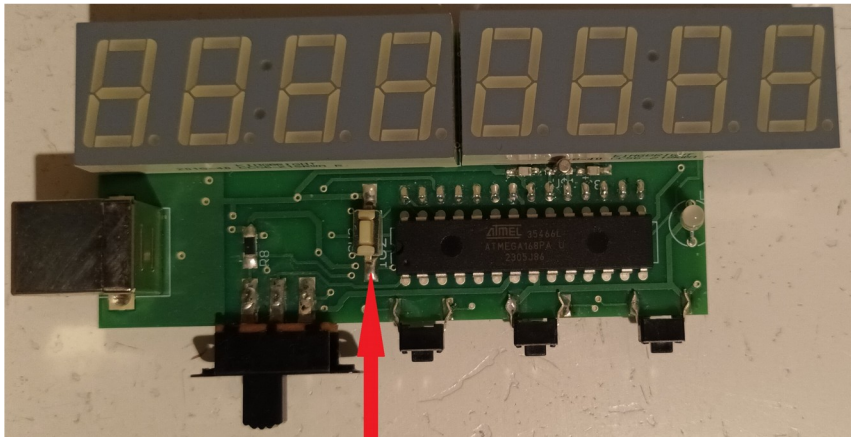
Note the locations of the following components

- The user switches sw1, sw2 and sw3
- The double pole double throw switch (DPDT)
- The reset switch

- The Atmega 168 which is in the top side of the pcb (note also the location of pin 1)
- The Atmega 328 which is in the under side of the pcb (note also the location of pin 1)

Because of the way the two Atmega devices are assembled the pins of one of them must be soldered from the device side of the board.

Note also the orientation of the two display modules.

DPDT switch sw1 sw2 sw3 	Photo of PCB-111000V2 under side. Solder the Atmega328 to this side carefully noting its placement and the location of pin 1.
	Photo of PCB-111000V2 top side. Solder the Atmega168 to this side carefully noting its placement and the location of pin 1. Note also the orientation of the display modules
Reset switch	