PCB_111000_V3: Getting a new pcb up and running

1. Introduction

PCB_111000_V3 is a development of PCB_111000_V2.
Both pcbs are loaded with two Atmega 328 devices rather than one Atmega328 and one Atmega168.

The first Atmega 328 hosts user projects (The user device).

The second 328 (the mini-OS device) hosts a mini-OS which:
Drives an 8 digit display
Includes an Atmega programmer to programs the user device.
Provides a variety of services for use by user projects

PCB initialisation is carried out using the following Arduino sketches

UNO_PCB_111000_V3_programmer	(was 8_UNO_AVR_programmer_V3)
mini_OS programmer	(was 5_Project_pcb_168_V2.30_Arduino_V2)

2. Setting up PCB-111000_V2: Basic steps

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

Step 2
Open the sketch "UNO_PCB_111000_V3_programmer". Under Tools/Board select Arduino UNO.
Connect the UNO to the PC check the port number and upload the sketch.

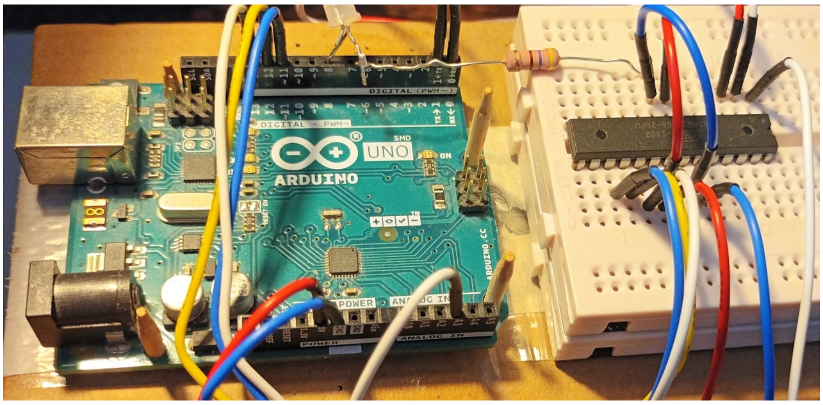
Step 3
Close the sketch assemble the HW (see section 3) and with the aid of the Br@y++ terminal emulator use the UNO to:
Upload "mini_OS programmer" to the user device
Calibrate it's internal RC oscillator
Allow the user device to run so that it can upload string data to its EEPROM.
Note:
Terminal emulator runs at 38.4KB and uses default settings.
See section 4 for more details.

Step 4 Load the user device onto the PCB111000_V3 and test, then add the mini-OS device.

Step 5
Run "mini_OS programmer" under the control of Br@y++ to setup the mini-OS device (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 "mini_OS programmer" 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 user device before it is soldered onto the PCB111000_V3

Note: Atmega328 pin numbers given for a DIP28 device

Upload "UNO_PCB_111000_V3_programmer" sketch before placing the user device

UNO			Plug-in pcb
UNO Pin	Atmega 328		User device
	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 "UNO_PCB_111000_V3_programmer"

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 328 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 "mini_OS programmer"

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 "mini_OS programmer_V1"

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 the user device 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 user device and note that it now offers a recalibration

5 Driving "mini_OS programmer"

5.1 Add the user device

Solder the User device to the PCB-111000_V3.
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 mini-OS device

Solder on the mini-OS device 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 hex file

Press -p- and send
PCB_111000_V3\mini-OS_HEX_files\mini_OS_CC_display.hex
When the LED stops flashing send pcb_bootloader_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 mini-OS device

Follow instructions to auto cal the mini-OS device and restore the "P/S……" prompt.

5.5 Upload the "Hello_world" file to the mini-OS device 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

6A To test programmer

Press the DPDT switch right to get the "p/g   p/g   p/g ….." (May need to press the pcb reset switch.)

Press -p- and send
PCB_111000_V3\setup_projects\First_test_proj\First_test_proj.ino.standard.hex

At the "p/g   p/g   p/g …..." prompt press -g- 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 mini-OS device is being programmed properly and the displays can be added.

6B To test the I2C bus

Send PCB_111000_V3\setup_projects\Get_mini_OS_version\Get_mini_OS_version.hex
Press x at the p/g prompt and then 3 to download hello world messages.

7. Load and check the displays

Displays are driven using a multiplexer which can make fault finding difficult.
Repeat as for section 6 but send "PCB_test\PCB_test.ino.standard.hex". 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/g p/g p/g p/g p/g…." user prompt
and send mini_OS programmer
Run the project to get the "P/S   P/S   P/S…………."
Press S and reload file "on-chip_strings.txt" containing programmer strings
Press P then upload the new mini-OS as described in section 5.3

Note: Remember to delete line :00000001FF from end of new mini-OS file

9. Optional system files

Mini-OS_Resources\Eeprom_subroutines_PRN_alt.c:

In early designs PRN generator write to the EEPROM was too frequent and EEPROM burnout occurred. In this file the PRN uses a different location. It should be used if the Message from the OS is always the same.

Bootloader_resources\Bootloader_main_with_OS_reset.c

This enables user projects to reset the mini-OS which is necessary if the I2C bus crashes.
It was optional for PCB_111000_V2 but is standard for PCB_111000_V3.
See section 14 for more details.

10 Default User Project

Program PCB_111000_V3 with “mini_OS_reset.ino.standard. hex”. At the p/g p/g prompt press -x- in place of -g-. This will give a series of options depending upon the project.

11 SW downloads

There are two essential software applications that must be downloaded onto this PC. They are:

The Arduino development environment. Go to https://www.arduino.cc/en/software and download it from the Microsoft store.

Bray++ version 20130820 which can be downloaded from https://www.sites.google.com/site/terminalbpp/

Almost as important are

The portable version of Programmers notepad which can be downloaded from http://www.pnotepad.org/download/.

An excellent text written by Joe Pardue that can be downloaded from https://epdf.pub/c-programming-for-microcontrollers.html

The Atmega data sheet that can be downloaded from https://www.docdroid.net/Q6jfzd3/atmega48a-pa-88a-pa-168a-pa-328-p-ds-ds40002061a-pdf

12 Getting the PCB manufacturing files

Open the board in Eagle fusion.
Click on the manufacturing tab.
Click on the Export Gerber NC drill……. manufacturing icon then OK

Save as a CamOutputs.zip file when requested
Can now extract files to a CamOutputs directory and remove the “assembly folder” leaving “DrillFiles” and “GerberFiles” before recompressing the directory.

Send this to a pcb manufacturer.

13. PCB assembly

Fusion/Eagle circuit design and layout drawings can be found under
PCB-111000_V3\Version 3 design together with a parts list.
The Eagle pcb design software is required to view these

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 user device which is in the top side of the pcb (note also the location of pin 1)
The Mini-OS device 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.

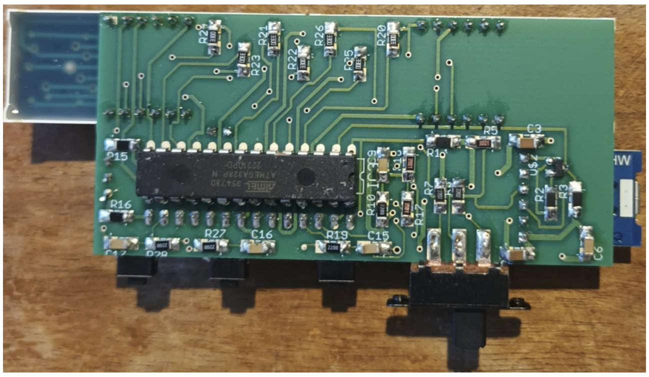


Photo of PCB-111000V3
track side.

Solder the mini-OS device to this side
Note its placement with pin 1 (towards the middle of the pcb).

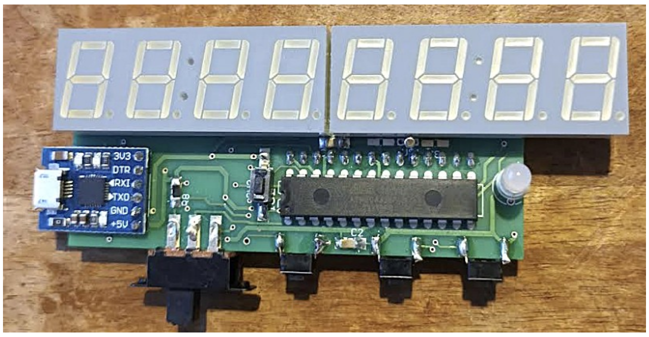


Photo of PCB-111000V3
component side.

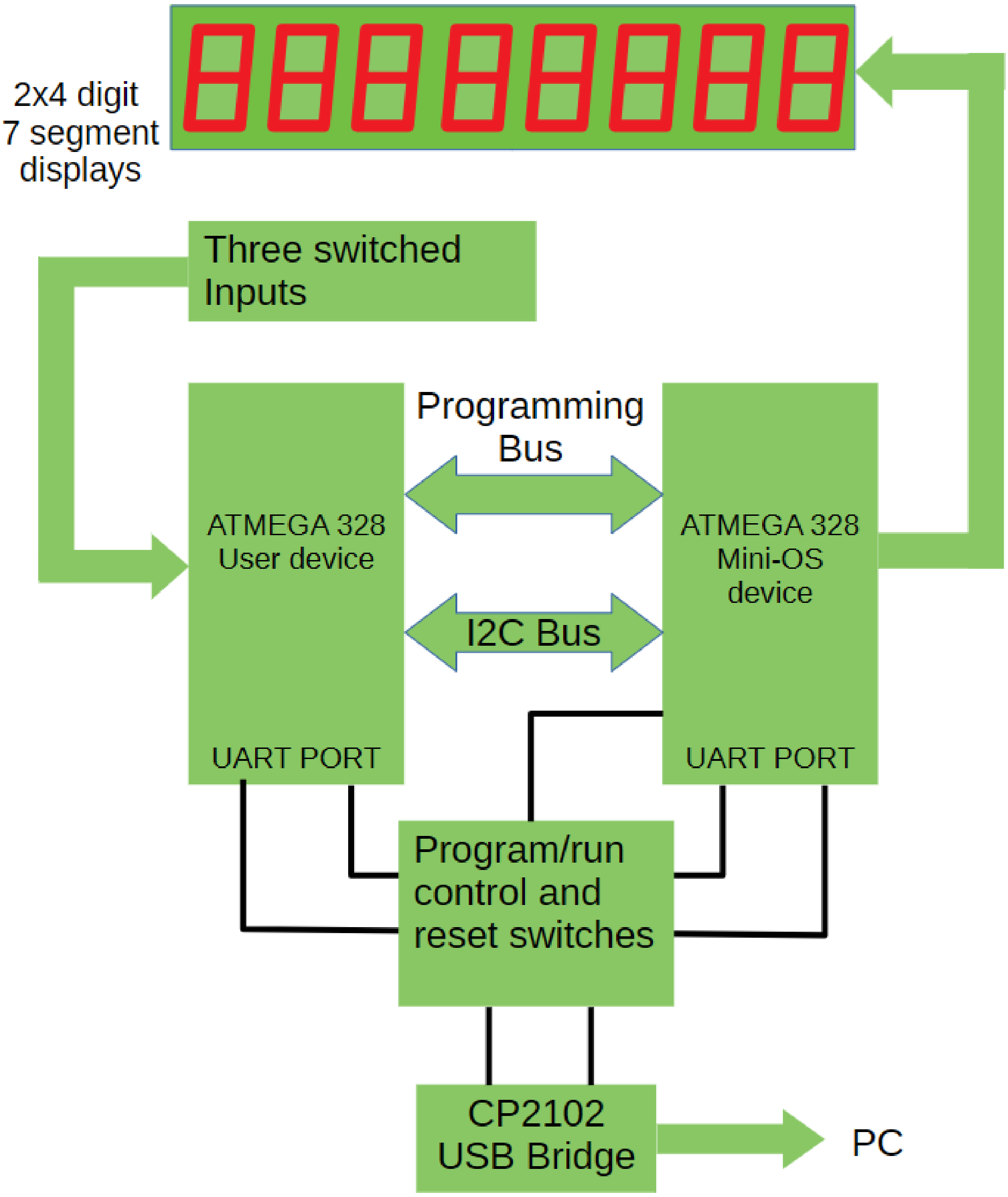
Solder the user device to this side
Note its placement with pin 1 (towards the middle of the pcb).

User switches left to right are sw1, 2 &3
Note also the orientation of the display modules

Other switches are the DPDT and the reset switch

14 System Block diagram

User device hosts user programs.
Mini-OS device hosts programmer and I2C/display……..application



15 Reset functions

There are three applications that may need to be reset:

The user project.
The programmer
The I2C/display driver

The reset switch

Resets the user project
Passes control in the mini-OS device to the programmer (i.e. a jump to the boot section).

Occasionally it is necessary to

Reset the user project
Pass control in the mini-OS device to the I2C/display code, (a jump to the application section).

This is referred to as a full reset and can be used when a project crashes and the reset switch fails to restart things as required.

To implement a full reset:

User code includes a watch dog time out with ISR.
Control jumps to the ISR when the I2C bus crashes.
This uses a programming wire to signal the mini-OS device and then
applies pulse to its reset pin.
Finally the ISR times out and the user device resets.
Note:

The ISR can also save the variables needed to ensure that the project resumes as required.

Full reset will only be applied to user projects when needed. As an example it has been implemented in

setup_projects\mini_OS_reset\mini_OS_reset.ino.

To indicate that the mini-OS has been implemented to accept full reset the user prompt is changed from

p/r p/r p/r……… to p/g p/g p/g……..

16 Issues

If -p- is pressed at the p/g prompt the first task of the programmer is to switch the target device into programming mode after which its programming memory is reset and the user is requested to send the code. Occasionally the device fails to enter programming mode. It is believed that this unpredictable response may be due to a dry joint in the reset circuitry.

Having pressed -p- at the p/g prompt the user device program space is cleared, in other words the original program can no longer function. In some versions of PCB 111000 this process was delayed until the first character of the new code had been received so that users could safely change their minds after pressing p. This has not been implemented here however as it was not found to be adequately reliable.

The original mini-OS was developed using winAVR will full optimisation, in other words to make the executable file as small as possible. winAVR has been replaced by Studio 7 which generates even smaller code. Optimisation has had to be turned off. The reasons for this have not been investigated but may be due to timing issues.