

## PCB111000\_1

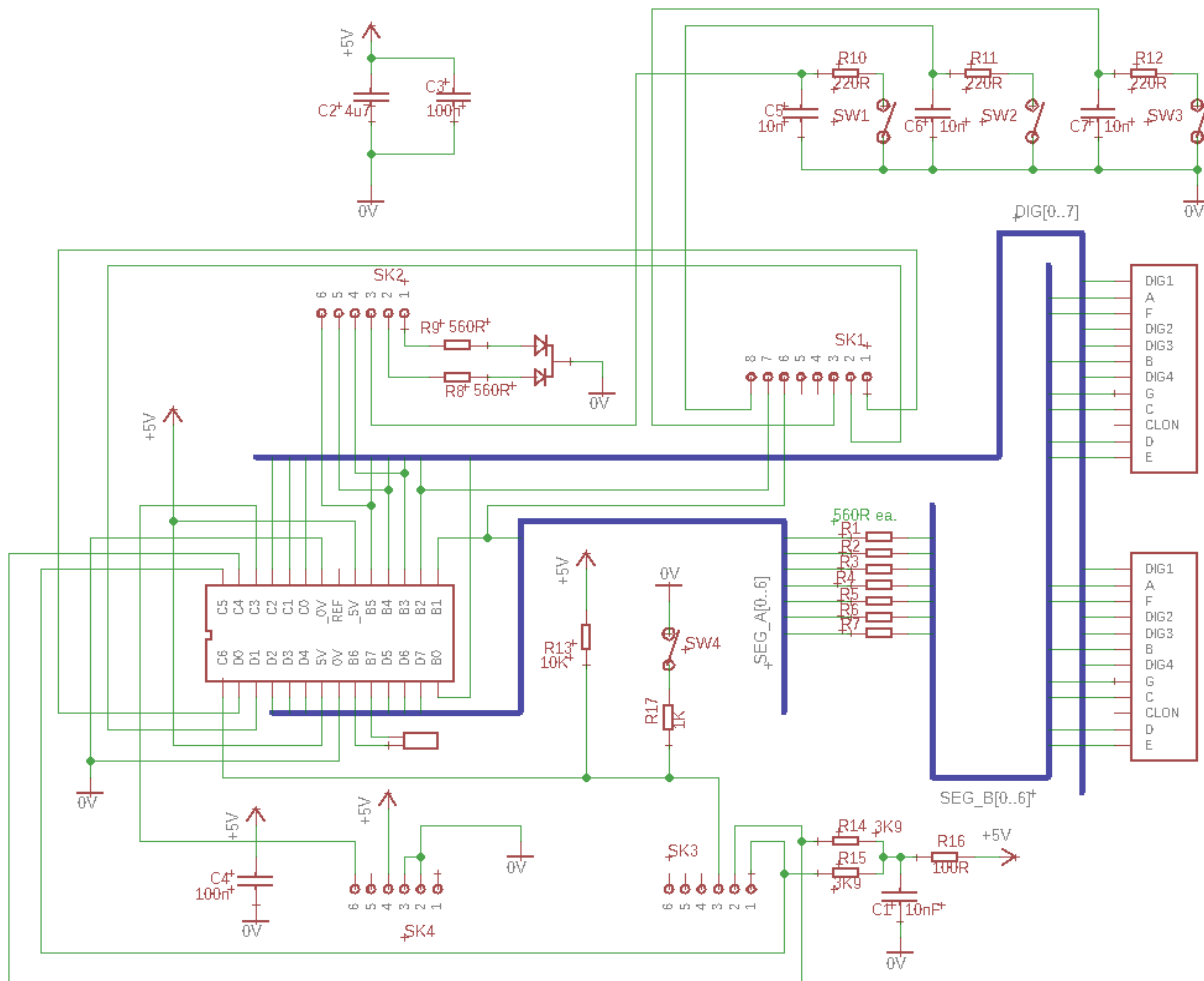
### What is it

It is a simplified version of PCB111000 designed to plug onto an Arduino UNO pcb. It contains the display, the OS device (an Atmega 328 loaded with the mini-OS), user switches and the dual led.

Note: The mini-OS consists of the display driver and PCB bootloader.

The UNO supplies the USB bridge, power and sockets and the user device (an Atmega 328 that hosts an Atmega bootloader and the user projects).

### Circuit diagram



#### Note:

A watch crystal is connected between B6 and B7

The diodes are LEDs

The sockets are used to connect PCB111000\_1 to the UNO pcb.

PCB111000\_1 is also referred to as PCB\_A

The combination of PCB\_A plugged into the UNO is referred to PCB111000\_UNO

Both PCB\_A and the UNO are loaded with Atmega 328 devices, however the UNO device is clocked using a 16MHz resonator and the PCB\_A device using its internal 8MHz oscillator. They therefore run under different sets of configuration bytes. This is used to distinguish between the UNO and PCB\_A when the flash/EEPROM of their devices is being programmed.

## Project system applications

UNO Hex/text bootloader. This resides in the boot partition of the user device and programmes its flash with hex and text files (it replaces the standard UNO bootloader).

Project programmer. This programs both EEPROM and flash (hex files only). When running on the UNO device it is used to program the pcb\_A device (with the mini-OS or a copy of itself). However when running on the pcb\_A device it can not be used to program the flash of the UNO device. This is done using the mini-OS. Two almost identical versions of the project programmer exist. “Project\_programmer\_AVR.c” is compiled using WinAVR and “Project\_programmer\_UNO.ino” is compiled using the Arduino development environment. Both versions are identical in all other respects.

Mini\_OS The display controller and I2C master reside in the application partition of the pcb\_A device. The PCB-bootloader resides in the boot partition of this device. Its main function is to overwrite the original UNO bootloader with the UNO\_Hex/text bootloader. After that it should only rarely be required.

## Why interfere with the Arduino code

The UNO comes with a very good bootloader. The idea here however is to combine the UNO with PCB 111000\_1 to create a new product k/a PCB 111000\_UNO that will replace the original PCB 111000 and be very easy to assemble.

The benefit of having access to programmer code is that each use of the programmer is tailored to an individual requirement. Hopefully this should enable PCB 111000\_UNO to perform as a single product rather than the combination of two quite different ones.

Two additional features are also offered: Text programming of the flash and optional verification. At the h/t/r prompt press h or t for programming without verification and H or T for programming with verification.

## EEPROM Reservations

	UNO device	PCB_A device
0x3FF	<i>User cal if set</i>	User cal if set
0x3FE	<i>User cal if set</i>	User cal if set
0x3FD	<i>Default cal supplied by Atmel</i>	Default cal supplied by Atmel
0x3FC	<i>Copy of MCUSR</i>	Diagnostic mode
0x3FB	<i>prog_counter_H</i>	Multiplexer period
0x3FA	<i>prog_counter_L</i>	<i>Not used</i>
0x3F9	<i>cmd_counter_H</i>	Reset status_1
0x3F8	<i>cmd_counter_L</i>	prog_counter_H
0x3F7	Reset control	prog_counter_L
0x3F6	PRN EEPROM reservation UNO_bootloader	cmd_counter_H
0x3F5	PRN EEPROM reservation UNO_bootloader	cmd_counter_L

0x3F4	Flash text string pointer	Reset_status_2
0x3F3	8 bit PRN gen user app	8 bit PRN gen EEP subroutines
0x3F2	16 bit PRN gen user app	
0x3F1	WDTout detected	Signals Auto cal

Note: reset status is written by the Project Programmer to indicate a device has just been programmed.

### ARDUINO UNO connections

Refer to circuit diagram at the beginning of this document.

SK1		SK2		SK3		SK4	
1	D0 RXD	1	B0	1	C5 SCL	1	
2	D1 TXD	2	B1	2	C4 SDA	2	0V
3	D2	3	B2	3	C3	3	0V
4	D3	4	B3 MOSI	4	C2	4	5V
5	D4	5	B4 MISO	5	C1	5	3V3
6	D5	6	B5 SCK	6	C0	6	C6 RESET
7	D6						
8	D7						

### PCB111000\_1/Arduino interconnections

Note:

The Arduino\_UNO (to be k/a the UNO) contains the user IC, the one that hosts user projects and Atmega bootloader. PCB111000\_1 (to be k/a pcb\_A) contains the IC that hosts the mini-OS.

	PCB111000_1	UNO	
	SW1	B2	User switches
	SW2	D7	
	SW3	D2	
	SW4 C6 RESET	C3	Resets mini-OS/programmer IC
	R8	B0	Dual LED driver
	R9	B1	
	R14	C4 SDA	I2C bus
	R15	C5 SCL	
	B3	B3 MOSI	Bi directional programming interface
	B4	B4 MISO	
	B5	B5 SCK	
	C3	C6 Reset	Resets the user IC
	B1	D5	Reset SW control for a CA display
	B2	D6	Reset SW control for a CC display

## PCB111000\_1 Display connections

	Segments	Port	Digits	Port
	A	B1	1	B5
	B	D2	2	C0
	C	D3	3	C1
	D	D4	4	C2
	E	D5	5	B0
	F	D6	6	B2
	G	D7	7	B3
			8	B4

### UNO device fuse settings (as delivered and as used in this project)

Extended	0x5	2.7V BOD	OK	
High	0xDE	as delivered	Change to 0xD0	
	1	Ext reset		
	1	Debug off		
	0	Serial programming		
	1	WDT under program control		
	1	Eeprom not saved at chip erase		Change to zero
	1	Minimum boot partition size		Change to zero
	1	Minimum boot partition size		Change to zero
	0	Reset vector to boot partition 0x7000		
Low	0xFF	16MHz low power resonator, 64mS SUT	OK	

### PCB\_A device fuse settings

Extended	0xFD (i.e. 0x5)	
High	0xD0	
Low	0xE2	64mS SUT internal 8MHz oscillator.

### Display segment currents

For 560R resistors the current per segment is 4.3mA (pd of 2.4V accros 560R)

For 220R resistors the current per segment is 9.1mA (pd of 2V accros 220R)

Consider the display 88888888:

The constant current in each 220R resistor is 9.1mA

The current to be sunk by the appropriate I/O pins is 64mA however the duty cycle is only 1 in 8 i.e. 12.5% and the mean current is mA well within the IC max limits.

Display absolute max dc current is 30mA

## TEST Procedure

### Setting up PCB 111000\_UNO

**Basic set up** Using Project\_programmer\_UNO only  
Start with a UNO pcb running the Arduino bootloader.

Open the Arduino GUI: Use it to upload Project\_programmer\_UNO onto the UNO device.

Open a terminal emulator: Run Project programmer\_UNO. Use it to program the PCB\_A device EEPROM with the Hello\_world file and flash with PCB\_A\_Mini\_OS\_I2C\_V17\_CC/CA.

Close the emulator: Upload Proj\_1A1\_UNO using the Arduino GUI. This clears Project\_programmer\_UNO from the UNO device and checks the operation of the display.

Re-open the emulator: Double click PCB\_A reset switch so that the display stops operating.  
Press key 'p' so that a '?' is displayed on the emulator.  
Then press key 'r' which starts a download of the Arduino bootloader from the UNO device.  
Save the bootloader for later use.  
Double click the reset switch again.  
This time press 'p' followed by 'p' and upload "UNO\_bootloader\_for\_hex&text\_V5".

The "h/r/t/D..." user prompt is automatically generated.

Use 'r' to run the default programs  
Press sw1 and pulse sw2 to read hello world strings  
Press sw3 to operate intensity control

Double click the reset switch  
Use 'h' and 't' to upload programs and their commentaries.  
Use 'r' to display the commentaries and run the programs  
r  
Power cycle and check that the user app starts up automatically.

Use 'D' to reinstate the default programs

Check that a single click of PCB\_A reset switch resets the user app. And a double click gives the "h/r/t/D..." user prompt

Having got the user prompt make keypress 'p' followed by 'q' and restore the original UNO bootloader to the UNO device. Upload Proj\_1A1\_UNO and check that it works correctly.

Done for PCB 111000\_UNO/CA  
Done for PCB 111000\_UNO/CC

/\*\*\*\*\*/

### Setup using a Project\_programmer on both devices

Start with the UNO pcb device running the Arduino bootloader.

Open the Arduino GUI: Use it to upload Project\_programmer\_UNO onto the UNO device.

Open a terminal emulator: Launch Project programmer\_UNO. Use it to program the PCB\_A device with Project programmer\_AVR

Note that Project programmer\_AVR running on PCB\_A launches automatically and the baud rate should immediately be reduced to 28800.

Program the UNO device EEPROM after which control automatically passes back to Project programmer\_UNO and a 56700 baud rate is required.

Pressing 's', 'x' reinstates Project programmer\_AVR running on PCB\_A and requires a 28800 baud rate.

Pressing 's', 'x' again reinstates Project programmer\_UNO and requires a 57600 baud rate.

As an alternative holding PCB\_A in reset when Project programmer\_AVR is running forces Project programmer\_UNO to run.

Attempt to program the UNO device flash. Note that this is not allowed (because of a config bytes incompatibility)

Launch Project programmer\_UNO, select 57600 baud rate press 's' 'p'. Program PCB\_A with PCB\_A\_Mini\_OS\_I2C\_V17\_CC/CA.

Upload Proj\_1A1\_UNO using the Auduino GUI. This clears Project\_programmer\_UNO from the UNO device and checks the operation of the display.

Double click PCB\_A reset switch and download the original Arduino bootloader code.

Double click PCB\_A reset switch and upload "UNO\_bootloader\_for\_hex&text\_V5".

Continue tests as for the basic setup.

Done for PCB 111000\_UNO/CA

Done for PCB 111000\_UNO/CC

/\*\*\*\*\*/

### **Upgrading firmware without recourse to the Arduino GUI**

Start with PCB\_A\_Mini\_OS\_I2C\_V17\_CA/CC and UNO\_bootloader\_for\_hex&text\_V5 running.

Put Project programmer\_AVR on both devices

At the h/t/r/D... prompt press "h" and put Project programmer\_AVR on the UNO device

At the h/t/r/D... prompt press "rsp" and put it on the pcb\_a device.

Reduce the bad rate to 28800 th restore the s...s.....s..... prompt

Press "s" to get 2ATMEGA (UNO)328P detected" message

Press 's' 'x' to swap between programs running on UNO and PCB\_A

or

When PCB\_A is running hold it in reset to force UNO to run

s

Put "hello world" strings on UNO.

Note: we are not able to program the UNO this way

With UNO running restore PCB\_A\_Mini\_OS\_I2C\_V17\_CA/CC to PCB\_A.

Double click PCB\_A reset switch then press 'p' 'p' and update the UNO SW.

Press 'r' and check that the default programs run OK.

Restore UNO OS and Upload Proj\_1A1\_UNO to restore normal operation.

Done for PCB 111000\_UNO/CA  
Done for PCB 111000\_UNO/CC

**Mistaken case** At the UNO\_update? Prompt program something like Proj\_1A1.  
It may or may not work and there will be no bootloader.  
Double click PCB\_A reset switch press p p and try the update again.

**Calamity case** At UNO\_update prompt send Project\_programmer so deleting the UNO bootloader.

Run the Project\_programmer on the UNO device and use it to put Project programmer\_AVR-rescue on pcb\_a. Use this to restore UNO\_bootloader\_for\_hex&text\_V5 to the UNO device.

Then

Pulse PCB\_A reset to get R prompt  
Press and hold it again to get h/t/r/D prompt  
press h and reload Project programmer\_AVR on the UNO device.  
Press r then s and release reset switch  
Reload PCB\_A mini-OS and get h/t/r/D.....user prompt and immediately press D

This will delete the Project\_programmer from the UNO device giving the default programs.  
T

**Second Calamity case** Its not possible to get the h/t/r/D.... prompt.

Happened after running "Proj\_9C\_ATMEGA328\_manual\_cal" but not sure why.  
Believe eeprom may have got corrupted.

"Proj\_9C\_ATMEGA328\_manual\_cal" exits with the following message  
Enter new user cal value or  
enter zero to delete the user cal or  
press the reset switch if OK.  
ALWAYS exit with the R...R...R prompt: press reset switch if necessary

Was a struggle to fix but eventually the following procedure did it.

Using a second UNO/pcb\_a pair  
Put "Project\_programmer\_AVR" on the UNO  
Put "Project\_programmer\_AVR\_rescue" on the pcb\_a  
At the reset prompt put "Proj\_1A\_AVR" on the UNO to get rid of "Project\_programmer\_AVR"

Take the failed UNO/pcb\_a pair and remove pcb\_a  
Replace it with the pcb\_a hosting "Project\_programmer\_AVR\_rescue"

At the R....R prompt press R and reprogram the UNO with “2\_UNO\_bootloader\_for\_hex&text\_V6”  
At the s...s....s prompt press s and reprogram pcb\_a with 1\_PCB\_A\_Mini\_OS\_I2C\_V18  
At the h/t/r/D prompt press t and program Proj\_1A commentary and hex.  
Press r to get the commentary  
Press X to run the program

Publicity

AVR freaks

<https://circuitdigest.com/editorial>

Arduino projects