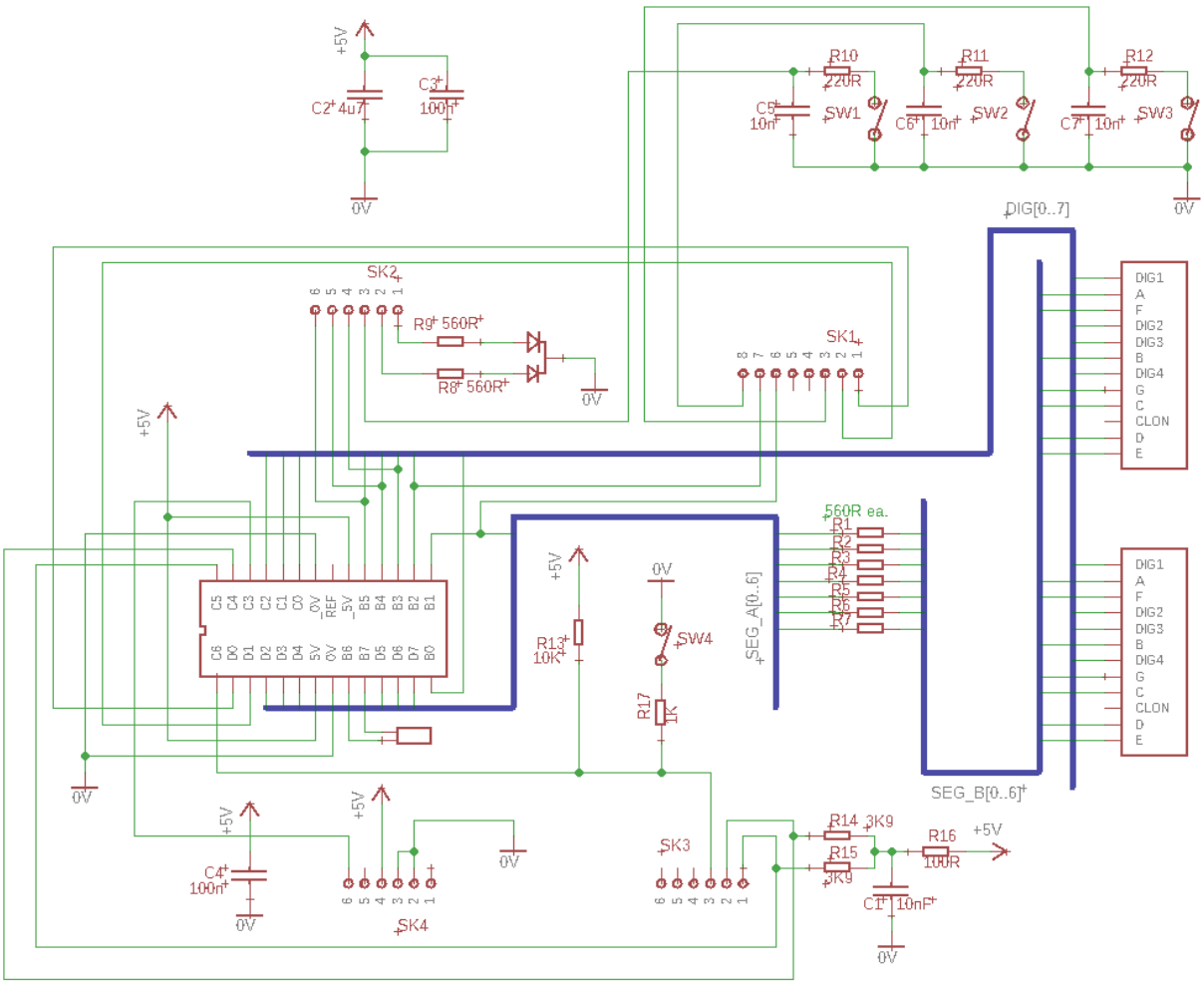
PCB111000_1

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.

PCB111000_UNO operation: Reset Options

Reset switches are provided on both the UNO and pcb_A.
Once the mini_OS and hex/text programmer have both been loaded the operation of the reset switches is as follows:

For the UNO
Reset takes program control to the "Hex/text bootloader" which generates the following user prompt
h/t/r h/t/r h/t/r h/t/r.....
The bootloader responds to keypresses as follows:
'h' A hex file is requested.
't' A text file is requested
'r' The hex file is allowed to run

For pcb_A
A single click of the reset switch takes program control to the display controller and the user application on the UNO device. (If no user application has been loaded the UNO must be reset.)

A double click of the rest switch results in the h/t/r h/t/r.....user prompt. This time however an additional keypress is available:

'p' If this is selected the pcb bootloader is launched.
This generates the following user prompt ? ? ?.....
which responds to keypresses as follows:
'p' The UNO device can be reprogrammed after which control passes back to the hex/text bootloader.
'r' Programming is aborted with no change to the UNO device.

At a power on reset the user application on the UNO and display driver of PCB_A both run so that user apps can run without access to a PC. (As is the case with the standard Arduino compiler.)

PCB111000_UNO operation: Issues

Three issues are noted as a result of which the system can latch up or one of the devices refuse to work as intended.

The main issue is that:
The TX port of the UNO and pcb_A are permanently connected together. (For PCB 111000 a DPDT switch was used to select with one would be connected to the USB bridge.) This means that care must be taken to ensure that the UART of only one device is ever active at any one time.

When user applications are running this is not a problem because the display driver does not use its UART. When the project programmer is running it can be a problem however especially if inappropriate code is loaded onto the pcb_A device. This is largely resolve by using a “weak reset”.
The user can hold the device in reset but the programmer can overcome this reset and remove the inappropriate code.

The second issue is that both UNO and pcb_A employ Atmega 328 devices. This issue has already been discussed above.

Finally, when using the system programmer or pcb bootloader, having pressed 'p' the target device is immediately erased. A modification ensures that it is not erased until the first character of the new hex file has been received.

As the result of these steps it is hoped that unintended operation, or corruption or loss of system code can be avoided. However in the last resort it will always be possible to reload the original Arduino code.

PCB111000_UNO operation: Setting up the UNO and pcb_A

As delivered the UNO already has a bootloader.
This is used to load the project programmer.
This in turn loads the mini-OS
The pcb bootloader is then launched and used to replace the UNO bootloader with the UNO_hex/text bootloader. (There will be no user prompt but a 'p' keypress will launch the pcb bootloader.)
User applications can now be launched.

PCB_A bootloader operation

All resets take program control to 0x7000, the start of the PCB_A bootloader.
For WDT, POR, BOR and immediately after programming, control jumps to zero (the start of the display driver).
Following external resets operation pauses for 30mS so that the PCB_A device can if required, be put into programming mode by the Project Programmer running on the UNO.

For a single click on the PCB_A reset switch

The signalling line is set low and held low for 250mS.
A 2mS pulse is generated which resets the UNO device
Following reset the UNO device reads its signalling line and jumps to the start of the user app.
A WDTout is generated after which control jumps to location zero (the start of the display driver).

For a double click on the PCB_A reset switch

The signalling line is set high
The UART Rx in enabled
A 2mS pulse is generated which resets the UNO device
The UNO device then generates the h/t/r user prompt
'h','t' or 'r' keypresses return control to the UNO (PCB_A executes a WDTout)
A 'p' keypress puts the UNO into programming mode and activates the UART Tx.
After programming PCB_A jumps to location zero and the UNO generates the h/t/r prompt.

UNO_bootloader operation

The UNO has a 16MHz crystal from which an 8MHz clock is generated using the system clock prescaler.

All resets take program control to 0x7000, the start of the UNO bootloader.
For external resets only, jump to the start of the calibration routine.
On return from this routine continue to the UNO bootloader
For all other resets jump to location zero (the start of the User App).
For external resets only next read the signalling line.
If it high (its default state) issue the h/t/r user prompt
If it is low issue a WDTout and jump to location zero (the start of the User App).

Project_Programmer

This can run on either device.
The system clock prescaler provides an 8MHz clock for the UNO and 4MHz for PCB_A.
Following a 5mS delay immediately after release from reset the programmer resets the target device.
Programming then proceeds as follows:

EEPROM of either device can be read or written to
Only the flash of PCB_A can be programmed
If this is done address 0x3F9 of its EEPROM is also set to zero.
The program aborts if an attempt is made to program the UNO flash.
(the devices have different configuration bytes).

Following programming of either flash or EEPROM the UART is disabled and the target released from reset.

Use of programmer to install mini-OS on PCB_A

The programmer is first put on the UNO at location zero
It then runs and is used to put the mini-OS on PCB_A
When this is complete
Reset the UNO using the UNO reset switch.
Double click on PCB_A
Press 'p' then 'p'
Reload UNO bootloader to remove programmer.

Operation of Full programmer when on both devices

The programmer can also be used to replicate itself on PCB_A.

It can then be used to program the UNO EEPROM but not its flash.
Control is automatically handed to the programmer on PCB_A
The baud rate must be reduced to 28.8kB
When this is complete
Press the reset switch on PCB_A
Change the baud rate to 57.6kB
Press ‘r’ to run the programmer on the UNO
The mini_OS can now be installed or re-installed on PCB_A.
Remove the programmer from the UNO as above.

Setting up a new PCB111000_UNO: Test regime VERSION_2

Step 1 Either take a new UNO, the device of which comes preloaded with the Arduino bootloader or use the USBasp programmer to restore the flash and configuration bytes of a UNO Atmega 328 to their as delivered condition.

Step 2 Use the Arduino OS to load the project programmer onto the UNO Atmega 328.
Plug PCB111000_1 into the UNO pcb and open a terminal program.
Use the Project Programmer to program the PC111000_1 EEPROM and the flash with a second copy of the programmer.

Step 3 Change the baud rate to 28.8K and program the EEPROM of the UNO device.
Confirm that the programmer will not program the UNO flash

Step 4 Restore the 57.6K baud rate.
Hold PCB111000_1 in reset (Note: this is only a weak reset)
Press ‘S’ then ‘P’ and then release the reset switch and programme the mini-OS.
Press UNO reset switch when requested and get S S S S user prompt

Step 5 Use Arduino to download Proj_1A (to overwrite the programmer) and check that the program runs automatically. (This also checks the operation of the mini-OS.)

Step 6 Reconnect the terminal program and double click PCB111000_1 reset switch
Press ‘p’ then ‘p’ and download the new UNO bootloader when requested.
Check that it runs automatically. (i.e. the h/t/t...h/t/r...h/t/r prompt is displayed)

Step 7 Press ‘H’ and download Proj_2G. Check that it verifies OK. Run the program checking single and double clicks of the PCB_A reset switch.

Step 8 Double click the reset switch press ‘h’ and download the “short_app”.
Press ‘T’ and download “Sample Text File” checking that it verifies OK
Press ‘t’ and download it again.
Press ‘r’ and check that the strings are readout as required.

Step 9 Download the full Atmega programmer onto the UNO device
Use it to replicate itself on the PCB_A device
Reduce the baud rate to 28.8K
Try to read the UNO EEPROM and note that it is empty

Hold in reset switch Press R, S, E, R, 2 and read PCB_A EEPROM
Note that the PCB_A EEPROM is still programmed and
control automatically passes back to the PCB_A program

Reprogram the UNO EEPROM
Control automatically passes back to the UNO program

Step 10 Reprogram the mini-OS
Reset the UNO using its reset switch and get the h/t/r h/t/r/…..user prompt
Double click the PCB_A reset switch and press ‘p’ ‘p’.
Reload the UNO OS
Press the UNO reset switch and note that the h/t/r prompt follows immediately
Reprogram Proj_2G
Press ‘r’ and check it runs

Step 11 Program Proj_1A
Press ‘r’ and check it runs
Power cycle PCB111000_UNO and check that Proj_1A runs automatically

Assembling the UNO_hex_text_bootloader.

This consists of 4 separate modules:

1. Calibrator
2. Hex_verification
3. Text_Verification
4. Programmer

It is only the programmer that requires interrupts. For the other modules the first 6 lines of their hex files (the interrupt vector table) are deleted. For example the calibrator starts at address 0x5FA0. The start of the sixth line has address 0x6000 and the jump is to address 0x6008. For all these modules the final two lines of the hex files must also be deleted.

For the programmer module the last but one line (jump to 0x7000) must be deleted. The four hex files can then be copied into a single hex file with the programmer at the end.

The Calibrator is present so that the bootloader can also be used on a modified PCB111000 in which the Atmega 328 is used for both devices.

EEPROM Reservations

	UNO device	PCB_A device
0x3FF	User cal if set	User cal if set
0x3FE	User cal if set	User cal if set
0x3FD	Default cal supplied by Atmel	Default cal supplied by Atmel
0x3FC	Copy of MCUSR	Diagnostic mode
0x3FB	prog_counter_H	Multiplexer period
0x3FA	prog_counter_L	Not used
0x3F9	cmd_counter_H/Reset status	Reset status
0x3F8	cmd_counter_L	Not used

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.