PCB 111000: Setting up a new project pcb

Step 1 Initial assembly

Assemble the pcb with all surface mount components, starting with the mini-MOSFET and then the FT230X USB transceiver. Finally add the USB socket and connect the pcb to a PC. The FTDI device will request drivers (via the internet or a disc). The pcb will then be assigned a "com port" number (watch "device manager" window for details). When this is complete load the switches and LED.

Step 2 Adding the Atmega 168 and 328.

Use the programmer pcb to program:

the flash (program memory) of an Atmega 168 with "1_ATMEGA_Programmer_V2.29A.hex"
and its EEPROM with "ATmega programmer_EEPROM strings.txt".

(see "Sys_documentation\5_The_programmer_PCB_V1" and also calibrate the device if necessary.)

Fit it to the upper side of the project pcb, closest to the pcb edge and with Pin 1 closest to the DPDT.
Connect the pcb to a PC and run a terminal program.
Switch the DPDT left and the Atmega 168 should generate the "User prompt P P P ….."
Press 'X' and the pcb should respond with the following message "Target not detected".
An unprogrammed Atmega 328 should now be fitted to the lower side of the pcb (Pin 1 closest to the DPDT) and the watch crystal should also be added.

Note: When using the Atmega 168 running "1_ATMEGA_Programmer_V2.29A" to program the Atmega 328 responding to the user prompt with:

'X' sets the reset vector to location 0x0000
'P' sets it to the start of the boot partition
'P' is only required when the mini-OS is being programmed

Step 3 Loading the Atmega 328 with the mini-OS

Switch the DTDP to the left.
At the "User Prompt" press 'P' three times.
When requested send file

"1_I2C_V12_0_CC_part_1.hex", then when the download stops send
"2_I2C_V12_0_CC_part_2.hex" and finally send
"3_AT_bootloader_V4_22_CC.hex"

Press 0 at the "intger(0-FF)?" prompt.

The programmer responds by verifying the number of instructions that have been programmed.
Additional data given includes:
The programmed configuration bytes of the Atmega 328 for which critical parameters are:

Reset vector: Start of the bootloader code.
External reset, serial programming, watchdog under program control
EEPROM preserved through “chip erase”
Use of internal 8MHz clock.
3.9V brown-out

The calibration bytes refer to the Atmega 168 programmer. If it has been calibrated by project SW then the calibration data is duplicated.

Step 4 Program the Atmega 328 EEPROM

At the “User Prompt.” press 'P', 'P', 'e', 'W'
At the “intger(0-FF)?” prompt press '0'.
Send text file “ATmega programmer_EEPROM strings.txt”.
Notice that the strings are loaded into locations 0x200 to 0x3F6.
Note: This area of the EEPROM is used to back-up the programmer strings required if the Atmega 168 is to host “AT_Programmer_V2.29A”.

Repeat the process but press 'P', 'P', 'E', 'W'
Send file “Hello_world.txt”
Note that the strings are stored between location 0x5 and 0x1FC (503 characters)
This file can be replaced by any similar file of size 507 characters or less which can also contain numbers (additional characters will be ignored).
The contents of the EEPROM can be verified by pressing PPER or PPer at the user prompt.

Step 5 Calibrating the Atmega 328

The mini-OS includes an auto-cal routine that runs when power is applied. However users can also run “Proj_9C_ATMEGA328_manual_cal” if desired.
To do this:
Switch the DPDT to the right and press the vertical push button switch.
Note the “p/r p/r p/r ….” user prompt. Press 'p'
Send file “Proj_9C.hex” when requested
Press 'r' and operate the DPDT switch
The manual calibration routine will now run.

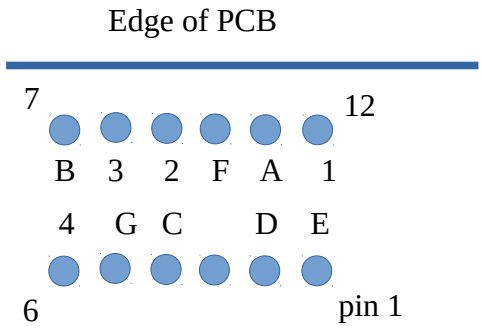
Step 6 Testing the project_pcb

At this stage the display modules can safely be fitted. It is often found that one or more of the display segments fails to work due to a dry joint or solder bridge. Fault finding is complicated because the display is driven by a multiplexer.

Project “Proj_9E_Test_PCB_Assembly” is therefore supplied to test the operation of the assembly. The multiplexer is disabled and digits and segments can be individually tested and faults located if any fail to work.

PCB 111000: Setting up a new project pcb

Driving the digits and segments



View of pcb from track side showing the pins of one of the four digit displays.

Letters refer to the individual segments.
Numbers refer to the individual digits.

Pin 3 is not used. It is provided to drive a decimal point or colon depending on which display module is used.

Pin-out of the display modules

Looking at the pcb from above the digits are numbered from left to right as follows:

Dig 7 Dig 6 Dig 5 Dig 4 Dig 3 Dig 2 Dig 1 Dig 0

Looking at the circuit diagram two displays are shown. Dig 4 to 7 are in the upper display and dig 0 to 3 in the lower display.

All digits are driven by the Atmega 328 which is on the lower surface of the pcb.

The Atmega 328 pins that drive the digits are as follows:

Pin 19 drives dig 7	Upper display pin 12
Pin 23 drives dig 6	Upper display pin 9
Pin 24 drives dig 5	Upper display pin 8
Pin 25 drives dig 4	Upper display pin 6
Pin 14 drives dig 3	Lower display pin 12
Pin 16 drives dig 2	Lower display pin 9
Pin 17 drives dig 1	Lower display pin 8
Pin 18 drives dig 0	Lower display pin 6

The segments are driven by the Atmega 328 as follows:

a	pin 15
b	4
c	5
d	6
e	11
f	11
g	13

PCB 111000: Setting up a new project pcb

Alternative method of setting up the project PCB firm ware

If the Atmega 168 is soldered onto the project pcb before its flash has been programmed:

Plug an Atmega 328 into the programming PCB and program
its program memory with “1_ATMEGA_Programmer_V1.08_Q.hex”.

(see “Sys_documentation\5_The_programmer_PCB_V1”)

Then press 'P', 'P', 'E', 'W' to program its EEPROM with “ATmega programmer_EEPROM strings.txt”.
Note: Text will be loaded into addresses 0x200 to 0x3F6.

Solder this IC onto the project PCB and use it to program the Atmega168 with
“1_ATMEGA_Programmer_V2.29A.hex” and then continue as above.

Note: Programming can also continue correctly even if the EEPROM of both devices are unprogrammed. However there will be almost no device generated prompts.