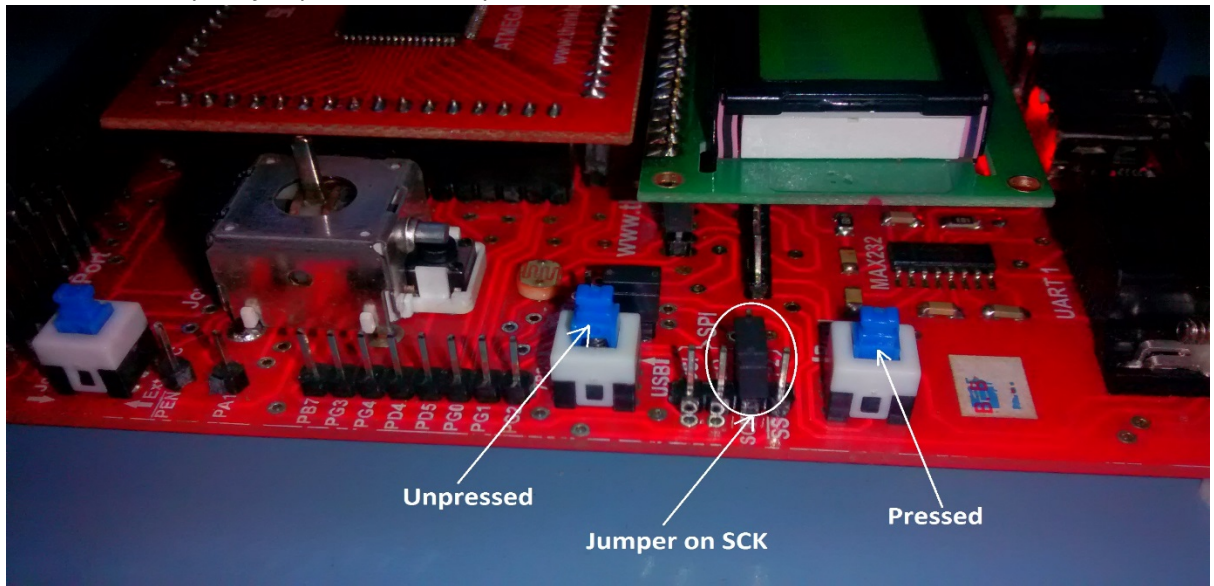
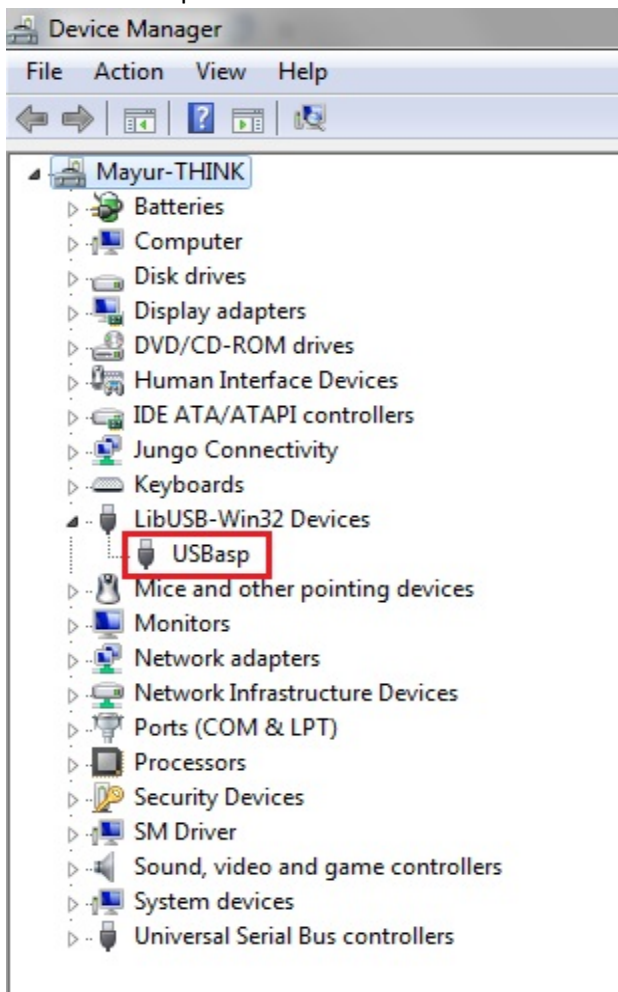


## UniBoard Getting Started Guide

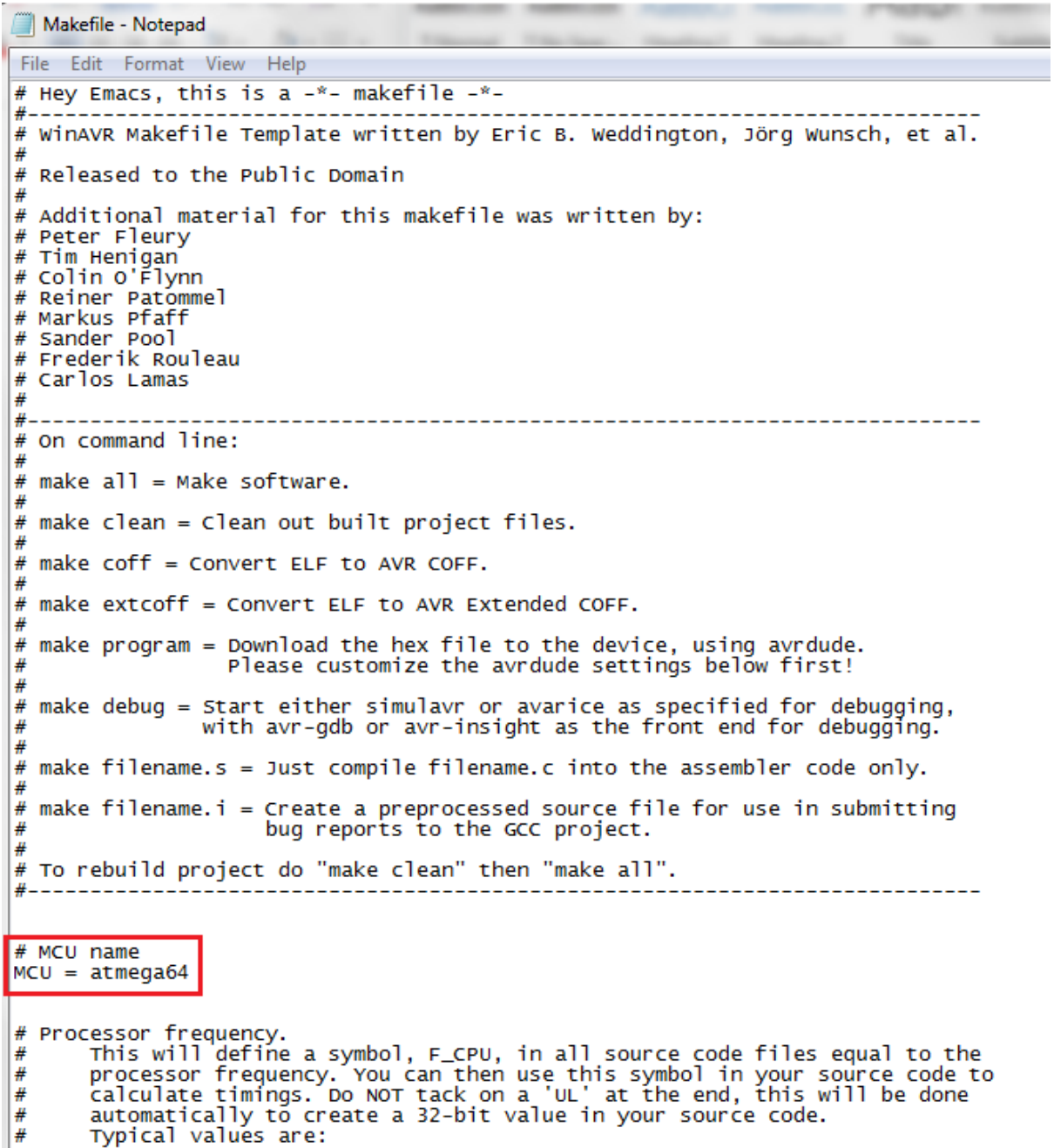
- 1) Check and verify the *jumper* and *switch* position as shown below:



- 2) Also check if 10-pin *FRC cable* from *PORTC* to *LED Port* is connected.
- 3) Now connect USB and check if board is powered ON or not.
- 4) Check if USBasp driver exists and it is detected in device manager



- 5) Download uniboard folder from FTP
- 6) Navigate to *uniboard\UNIBOARD\_SAMPLE\_CODES\non\_rtos\1\_ports*
- 7) Open Makefile in notepad and edit MCU as per your hardware (atmega128/atmega64)



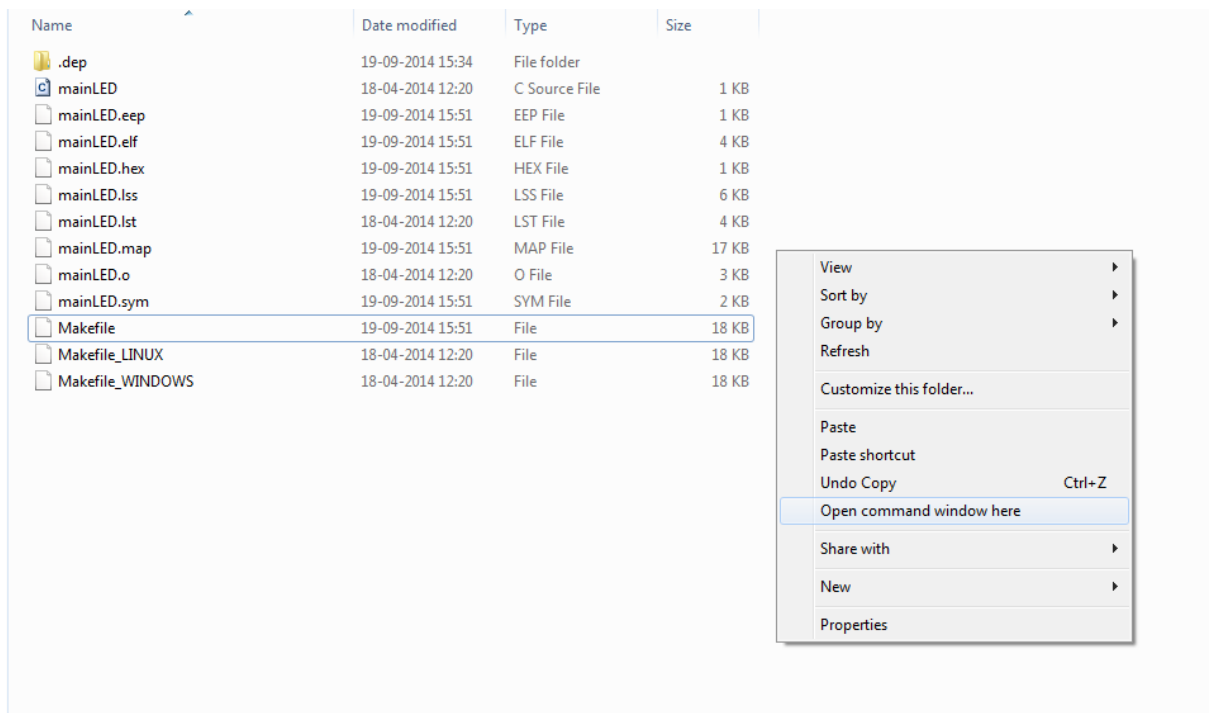
```

Makefile - Notepad
File Edit Format View Help
# Hey Emacs, this is a -*- makefile -*-
#-----
# winAVR Makefile Template written by Eric B. weddington, Jörg Wunsch, et al.
#
# Released to the Public Domain
#
# Additional material for this makefile was written by:
# Peter Fleury
# Tim Henigan
# Colin O'Flynn
# Reiner Patommel
# Markus Pfaff
# Sander Pool
# Frederik Rouleau
# Carlos Lamas
#
#-----
# On command line:
#
# make all = Make software.
#
# make clean = clean out built project files.
#
# make coff = Convert ELF to AVR COFF.
#
# make extcoff = Convert ELF to AVR Extended COFF.
#
# make program = Download the hex file to the device, using avrdude.
#                 Please customize the avrdude settings below first!
#
# make debug = Start either simulavr or avarice as specified for debugging,
#              with avr-gdb or avr-insight as the front end for debugging.
#
# make filename.s = Just compile filename.c into the assembler code only.
#
# make filename.i = Create a preprocessed source file for use in submitting
#                  bug reports to the GCC project.
#
# To rebuild project do "make clean" then "make all".
#-----
# MCU name
MCU = atmega64

# Processor frequency.
# This will define a symbol, F_CPU, in all source code files equal to the
# processor frequency. You can then use this symbol in your source code to
# calculate timings. Do NOT tack on a 'UL' at the end, this will be done
# automatically to create a 32-bit value in your source code.
# Typical values are:

```

- 8) Save and close notepad
- 9) while holding *Shift* key, right click on explorer and click on open command window here.



10) now enter following commands one by one.

- Make clean
- Make all
- Make program

11) If everything goes right, you will get following output.

```
E:\Uniboard\UNIBOARD_SAMPLE_CODES\non_rtos\1_ports>make program
avrdude -p atmega64 -P usb -c usbsp -U flash:w:mainLED.hex

avrdude: warning: cannot set sck period. please check for usbsp firmware update
avrdude: AVR device initialized and ready to accept instructions

Reading : ##### : 100% 0.02s

avrdude: Device signature = 0x1e9602
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed

        To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: warning: cannot set sck period. please check for usbsp firmware update
avrdude: reading input file "mainLED.hex"
avrdude: input file mainLED.hex auto detected as Intel Hex
avrdude: writing flash (242 bytes):

Writing : ##### : 100% 0.11s

avrdude: 242 bytes of flash written
avrdude: verifying flash memory against mainLED.hex:
avrdude: load data flash data from input file mainLED.hex:
avrdude: input file mainLED.hex auto detected as Intel Hex
avrdude: input file mainLED.hex contains 242 bytes
avrdude: reading on-chip flash data:

Reading : ##### : 100% 0.08s

avrdude: verifying ...
avrdude: 242 bytes of flash verified

avrdude: safemode: Fuses OK
avrdude done. Thank you.
```

- 12) The LEDs should blink now. If not, check the FRC cable and command prompt output for any errors.
- 13) In this way you can try other sample codes also. Try LCD code and you will see some output on LCD.
- 14) Now try for UART code.
- 15) Connect serial cable to your PC.
- 16) Open hyper terminal application.
- 17) Select bit per second as 9600, Data bits as 8, Parity as None, Stop bits 2 and Flow control as None and Click 'Apply' and 'Ok'
- 18) Press reset switch on board and then you will see a message printed on hyper terminal. Follow instructions on hyper terminal.
- 19) If all these things work, then you can conclude that your kit is working properly.

- Feel free to contact me if any problem persists.

- Happy Coding!