ComAidSystem 1.0 Makefile instructions

There are a few subtleties that make this makefile's configuration different from the standard makefile.

AVR parameters

a special compiler is used that is included with the AVR development kit. This requires a flag to specify the target micro controller. The following argument tells the compiler that our target is a ATMEGA168 chip.

-mmcu=atmega168

Makefile Chaining

We use 2 makefiles to build our project. One builds the required libraries and the other builds the project. The master buildfile 'Makefile' will call 'Makefile2' during the build.

Code Downloading

The code needs to be downloaded to the AVR device via a com port. The utility is called AVRDUDE which stands for AVR Downloader/UploaDEr. More information can be found here <http://www.nongnu.org/avrdude/user-manual>

-c {programmer-id} we use avr109 for our Atmel AppNote AVR109 Boot Loader

-p {part number} we use m168 for our ATMEGA168

-b {baudrate} we use 115200

-P {port} our development environment used COM5 but this will vary for every setup

Running

To build and download the code to the device simply run 'make' from the root of the source code. You will need to change the -P port setting to wherever the AVR programmer is connected. You will also need to ensure that the device is in program mode by switching the switch inside and restarting the device. The screen should be half filled with blocks if it is in the correct mode.

```
=Makefile============================================
GCCFLAGS=-g -Os -Wall -mmcu=atmega168
LINKFLAGS=-Wl,-u,vfprintf -lprintf_flt -Wl,-u,vfscanf -lscanf_flt -lm
AVRDUDEFLAGS=-c avr109 -p m168 -b 115200 -P /COM5
LINKOBJECTS= delay.o lcd.o uart.o nlcd.o

all:    comaidsystem-upload

comaidsystem.hex:    comaidsystem.c keyboard1.h
   make -f Makefile2
   avr-gcc ${GCCFLAGS} ${LINKFLAGS} -o comaidsystem.o comaidsystem.c
${LINKOBJECTS}
   avr-objcopy -j .text -O ihex comaidsystem.o comaidsystem.hex

comaidsystem.ass:    comaidsystem.hex
   avr-objdump -S -d comaidsystem.o > comaidsystem.ass

comaidsystem-upload:    comaidsystem.hex
   avrdude ${AVRDUDEFLAGS} -U flash:w:comaidsystem.hex:a

clean:
   rm *.o *.hex
=====================================================

==Makefile2==========================================
GCCFLAGS=-g -Os -Wall -mmcu=atmega168

all: comaidsystem.o delay.o lcd.o uart.o nlcd.o

comaidsystem.o: comaidsystem.c keyboard1.h
   avr-gcc ${GCCFLAGS} -o comaidsystem.o -c comaidsystem.c

delay.o: delay.c
   avr-gcc ${GCCFLAGS} -o delay.o -c delay.c

lcd.o: lcd.c
   avr-gcc ${GCCFLAGS} -o lcd.o -c lcd.c

uart.o: uart.c
   avr-gcc ${GCCFLAGS} -o uart.o -c uart.c

nlcd.o: nlcd.c
   avr-gcc ${GCCFLAGS} -o nlcd.o -c nlcd.c
=====================================================
```