How the Make files work

If I open up any of the make files at the top there a few standard lines about general settings
like the name of the project.
the chip being used
the CPU

```
PROJECT=hello.stepper.44.wave
SOURCES=$(PROJECT).c
MMCU=attiny44
F_CPU = 8000000
```

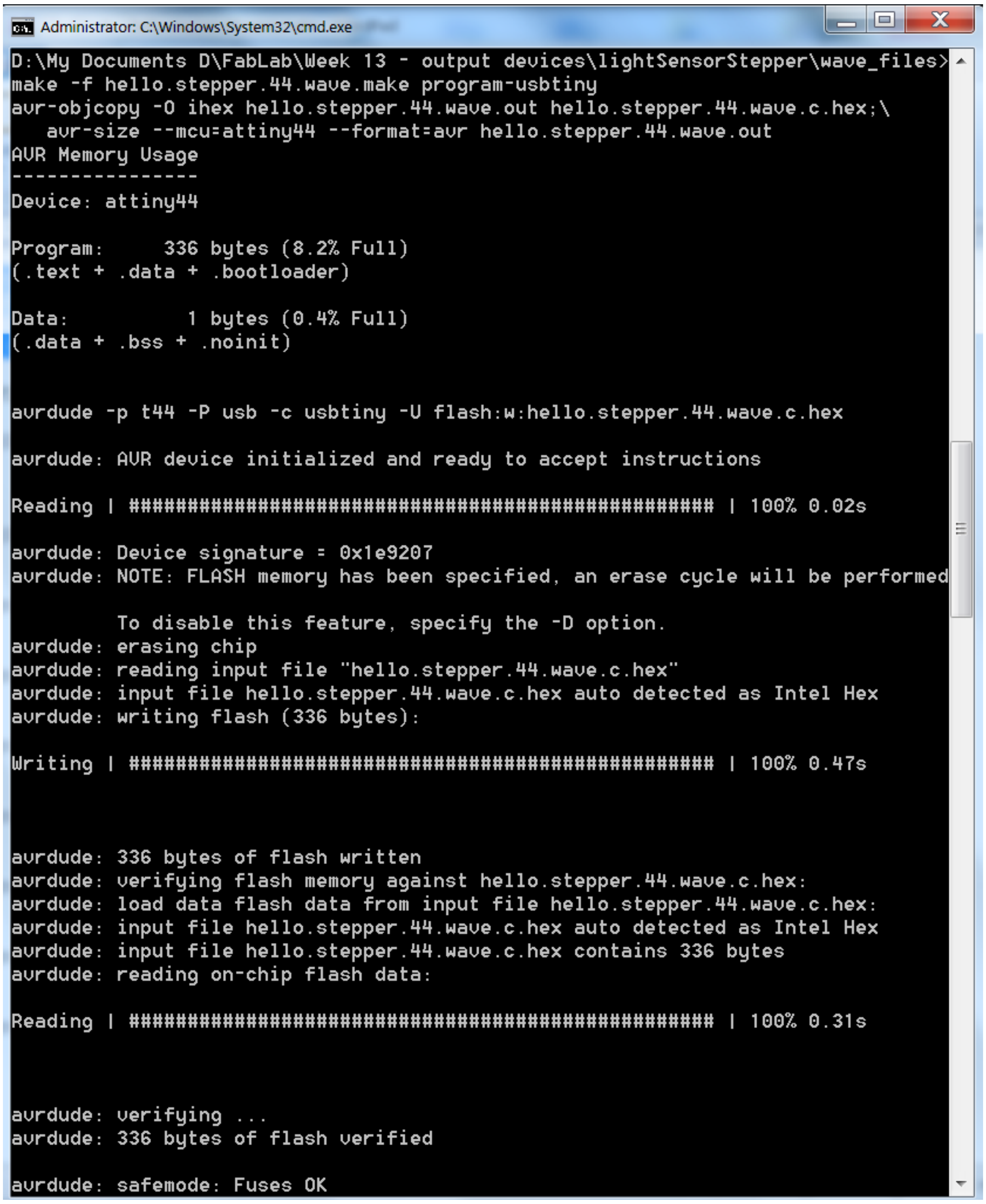
Then the code where there are the dollar signs are about what it needs to work. Anything after the semi-colon is what it needs to do

After this there are a number of commands you can get it to do. There are all listed in the bottom of the make file such as
program-usbtiny

if we look at the command line we can see if going off and doing these steps
it looks up everything is needs to have to carry out the requested command

- first checks the programme
- then checks for hex file
- then compiles the c

when it gets to compiling the c then it goes back up through these 2 things doing them one by one. It does this because first of all it checks if it can find the files or resources it needs and when it's sure it has all of them then it uses them to carry out the requested command.



```
D:\My Documents D\FabLab\Week 13 - output devices\lightSensorStepper\wave_files>
make -f hello.stepper.44.wave.make program-usbtiny
avr-objcopy -O ihex hello.stepper.44.wave.out hello.stepper.44.wave.c.hex;\
   avr-size --mcu=attiny44 --format=avr hello.stepper.44.wave.out
AVR Memory Usage
----------------
Device: attiny44

Program:     336 bytes (8.2% Full)
(.text + .data + .bootloader)

Data:          1 bytes (0.4% Full)
(.data + .bss + .noinit)


avrdude -p t44 -P usb -c usbtiny -U flash:w:hello.stepper.44.wave.c.hex

avrdude: AVR device initialized and ready to accept instructions

Reading | ################################################## | 100% 0.02s

avrdude: Device signature = 0x1e9207
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: reading input file "hello.stepper.44.wave.c.hex"
avrdude: input file hello.stepper.44.wave.c.hex auto detected as Intel Hex
avrdude: writing flash (336 bytes):

Writing | ################################################## | 100% 0.47s



avrdude: 336 bytes of flash written
avrdude: verifying flash memory against hello.stepper.44.wave.c.hex:
avrdude: load data flash data from input file hello.stepper.44.wave.c.hex:
avrdude: input file hello.stepper.44.wave.c.hex auto detected as Intel Hex
avrdude: input file hello.stepper.44.wave.c.hex contains 336 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 0.31s



avrdude: verifying ...
avrdude: 336 bytes of flash verified

avrdude: safemode: Fuses OK
```