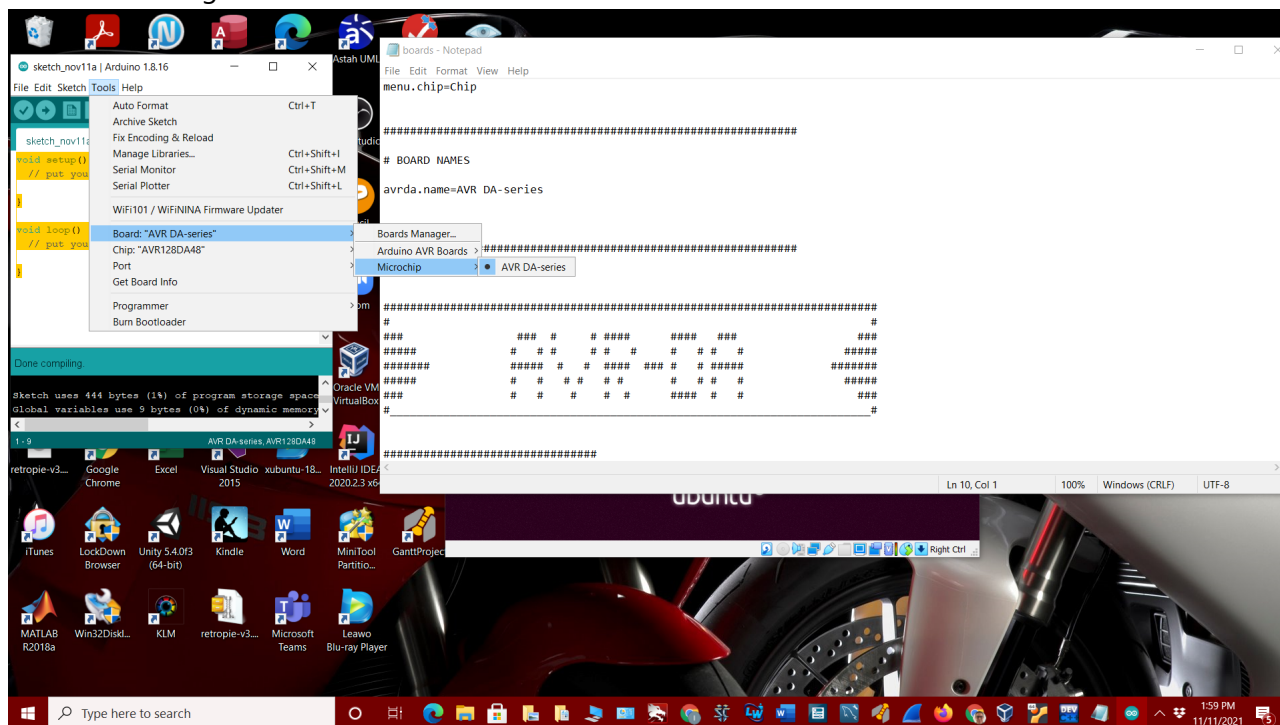


Start by creating a .txt file which will be basis for the various boards and their options for the Arduino IDE Core being created. The boards and configurations written in the file can be displayed under the Tools menu option within the Arduino IDE. Other options can be configured within the file as default and left out of the menu system of the Arduino IDE. To follow is a Boards.txt file for the Microchip AVR128DA48 board.

1. Create the menu option named Chip.

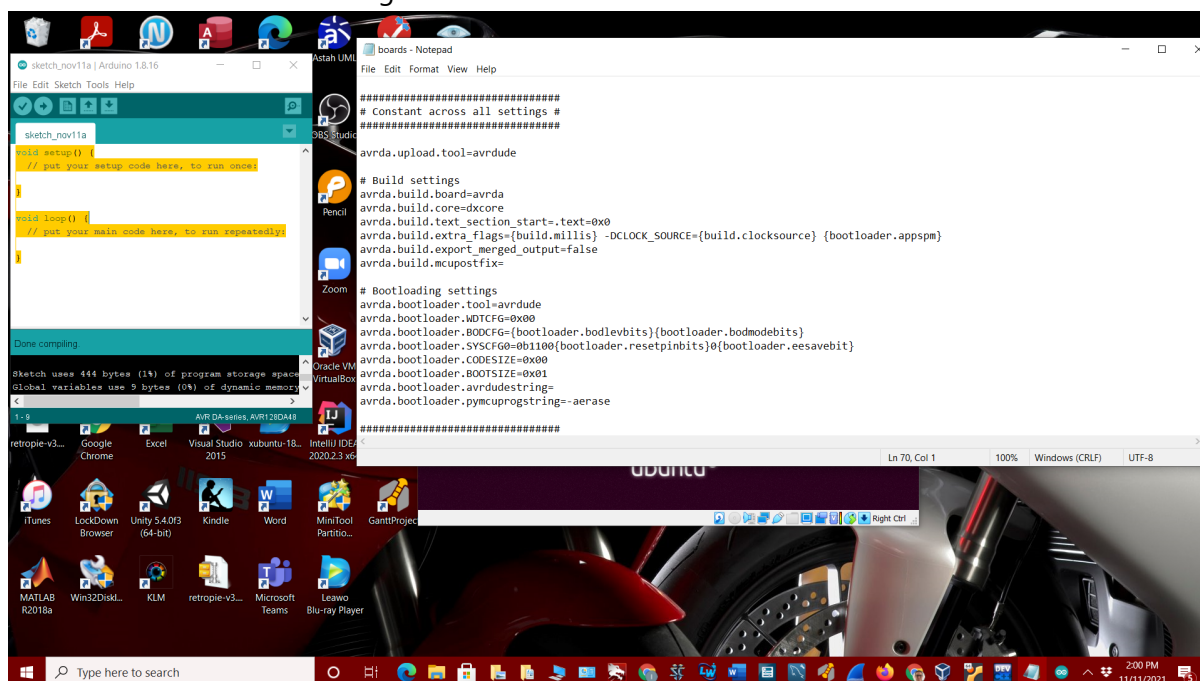
-
- The screenshot shows a Windows 10 desktop environment. The primary application is the Arduino IDE, which is open to a sketch named 'sketch_nov11a'. The IDE's 'Tools' menu is open, displaying options such as 'Auto Format', 'Burn Bootloader', and 'Programmer'. The sketch code is visible in the main editor, featuring comments in Indonesian and a reference to a GitHub repository for Arduino IDE-1.5-3rd-party-Hardware-specification. The desktop background is a dark image of a motorcycle. The taskbar at the bottom shows various application icons, including iTunes, Google Chrome, Visual Studio, and several utility tools. The system tray in the bottom right corner displays the date and time as 11/11/2021, 1:58 PM.

2. Create a heading named Board Names and add the desired boards.



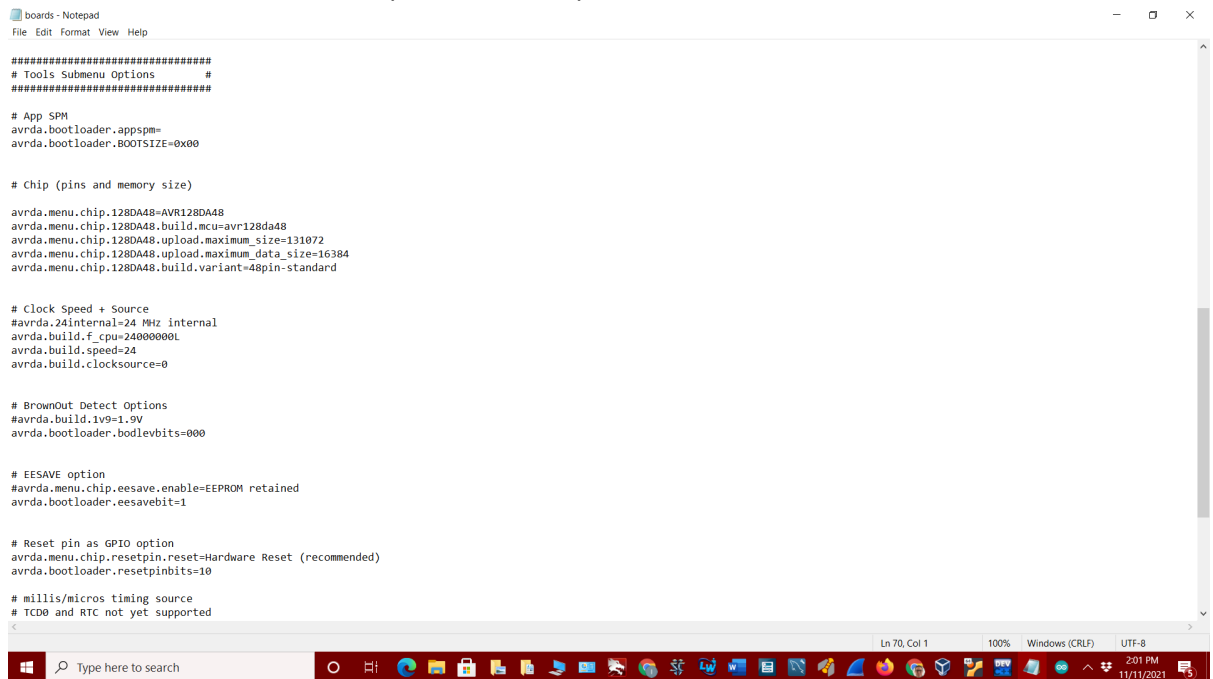
3. Create and set build settings and bootloading settings for your desired board(s).

- This will create constant settings for the board.



4. Create and set Tools submenu options that are default for the specific board.

- This will set default submenu options for the specified board.



```
boards - Notepad
File Edit Format View Help

#####
# Tools Submenu Options #
#####

# App SPM
avrda.bootloader.appspm=
avrda.bootloader.BOOTSIZE=0x00

# Chip (pins and memory size)

avrda.menu.chip.128DA48~AVR128DA48
avrda.menu.chip.128DA48.build.mcu=avr128da48
avrda.menu.chip.128DA48.upload.maximum_size=131072
avrda.menu.chip.128DA48.upload.maximum_data_size=16384
avrda.menu.chip.128DA48.build.variant=48pin-standard

# Clock Speed + Source
#avrda.24internal=24 MHz internal
avrda.build.f_cpu=24000000L
avrda.build.speed=24
avrda.build.clocksource=0

# BrownOut Detect Options
#avrda.build.lv9=1.9V
avrda.bootloader.bodlevbits=000

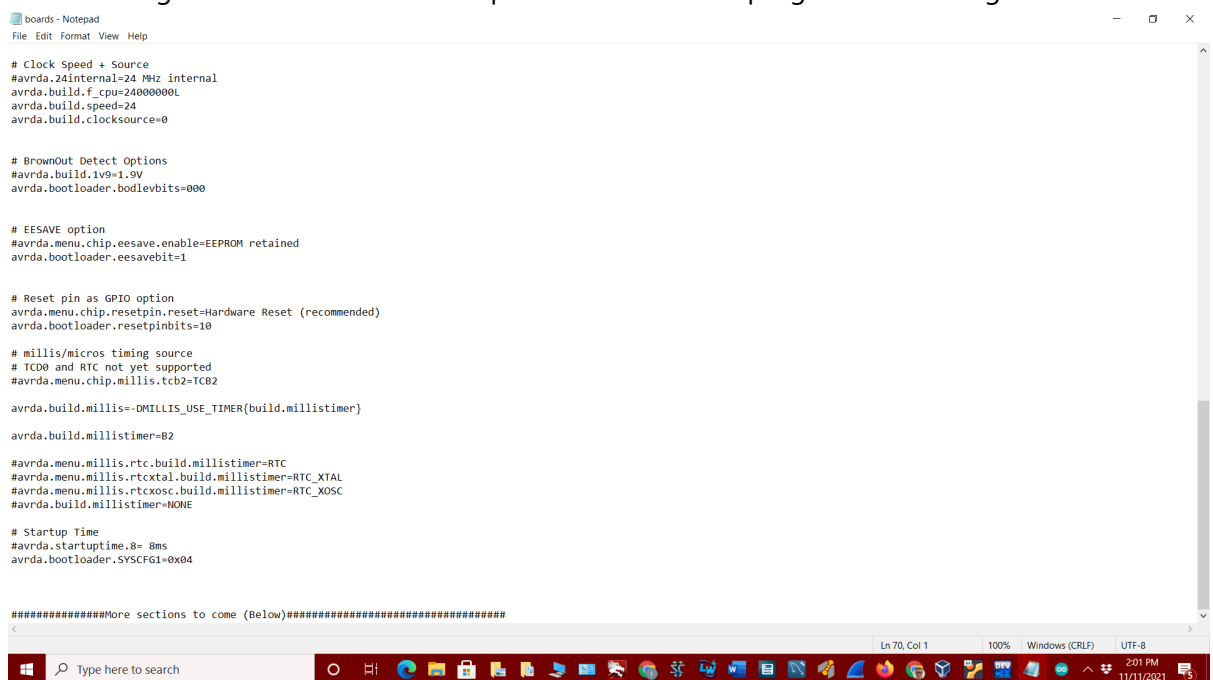
# EESAVE option
#avrda.menu.chip.eesave.enable=EEPROM retained
avrda.bootloader.eesavebit=1

# Reset pin as GPIO option
avrda.menu.chip.resetpin.reset=Hardware Reset (recommended)
avrda.bootloader.resetpinbits=10

# millis/micros timing source
# TCD0 and RTC not yet supported
```

5. Create and set other default options for the specified board.

- These settings will make it so that the specific board can be programmed through the IDE.



```
boards - Notepad
File Edit Format View Help

# Clock Speed + Source
#avrda.24internal=24 MHz internal
avrda.build.f_cpu=24000000L
avrda.build.speed=24
avrda.build.clocksource=0

# BrownOut Detect Options
#avrda.build.lv9=1.9V
avrda.bootloader.bodlevbits=000

# EESAVE option
#avrda.menu.chip.eesave.enable=EEPROM retained
avrda.bootloader.eesavebit=1

# Reset pin as GPIO option
avrda.menu.chip.resetpin.reset=Hardware Reset (recommended)
avrda.bootloader.resetpinbits=10

# millis/micros timing source
# TCD0 and RTC not yet supported
#avrda.menu.chip.millis.tcb2=TCB2

avrda.build.millis=--DMILLIS_USE_TIMER(build.millistimer)

avrda.build.millistimer=B2

#avrda.menu.millis rtc.build.millistimer=RTC
#avrda.menu.millis.rtcxtal.build.millistimer=RTC_XTAL
#avrda.menu.millis.rtcxosc.build.millistimer=RTC_XOSC
#avrda.build.millistimer=NONE

# Startup Time
#avrda.startuptime.8= 8ms
avrda.bootloader.SYSCFG1=0x04

#####More sections to come (Below)#####
```