## Step 1

Insert the programming socket with the pins facing upwards in the back of the board .

Connect the programming board to the pc using a usn mini cable


## step 2

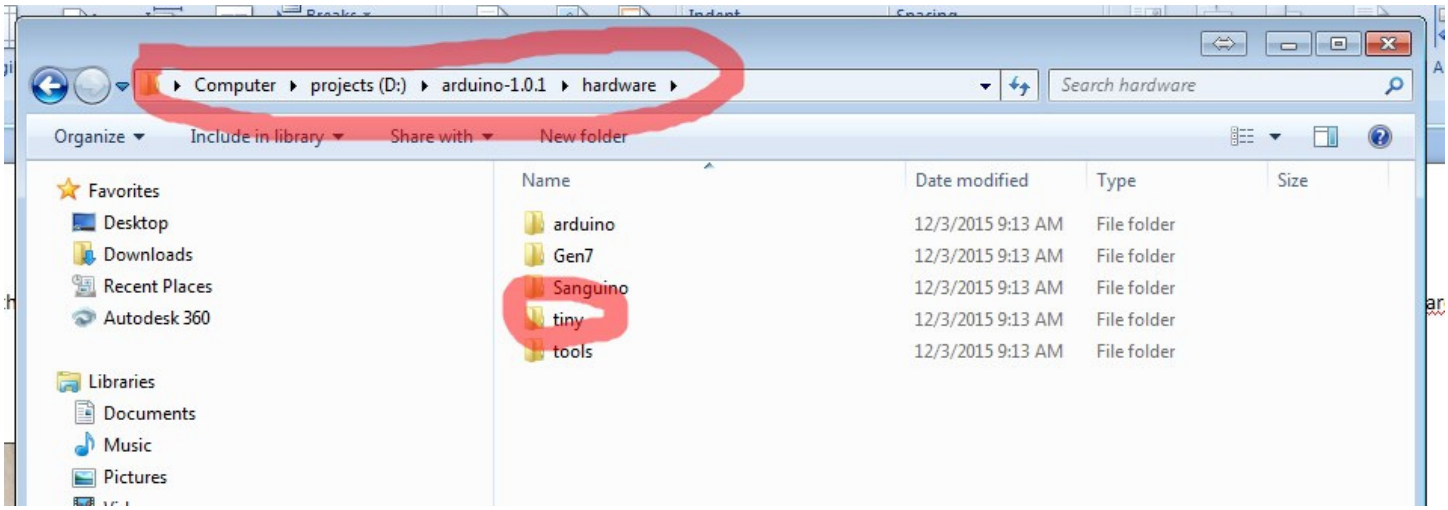
download arduino 1.0.1 from the arduino site ( old versions ) the newer ones have a problem

## step 3

download the Arduino sketch and the tiny library from the github

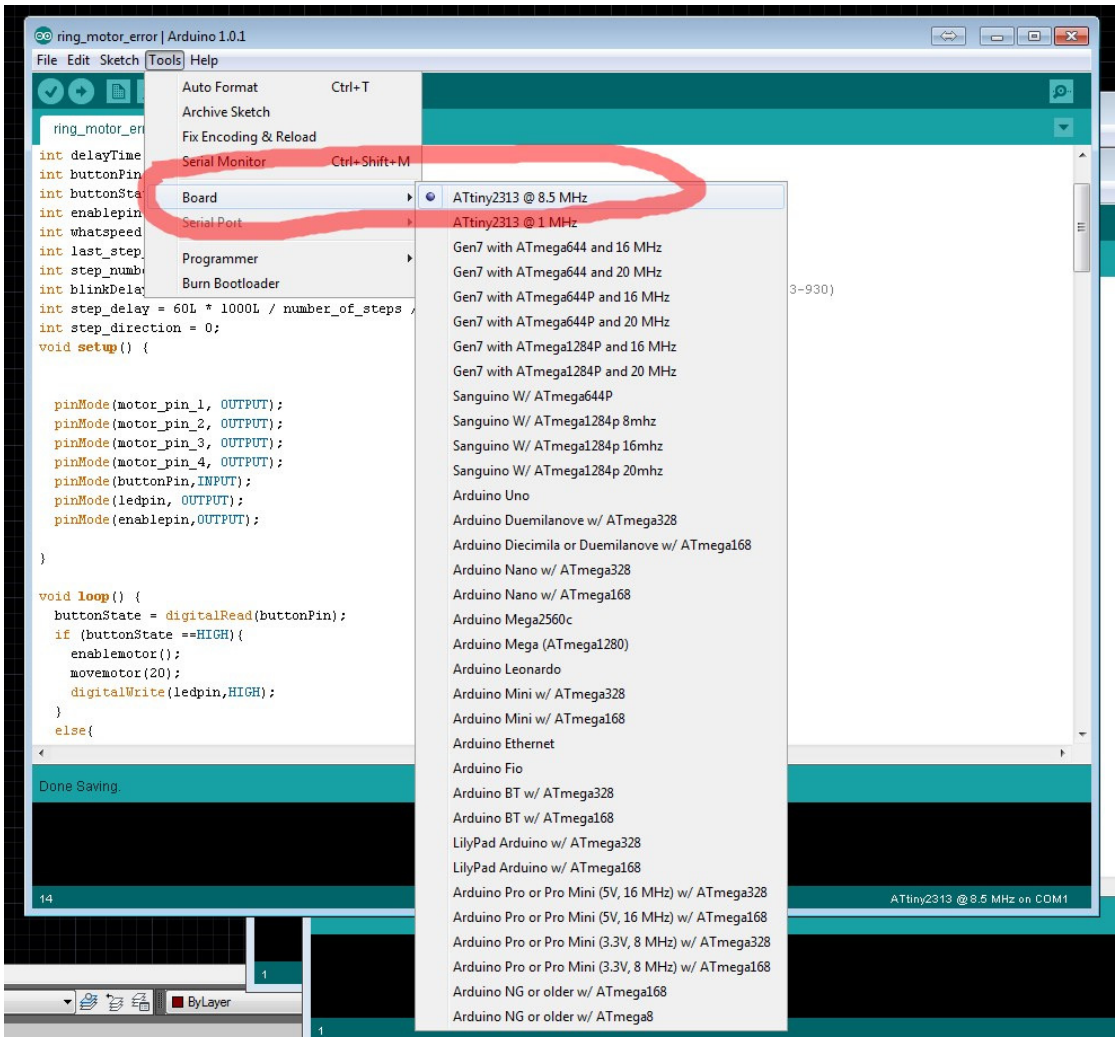
https://github.com/ekaggrat/holo_clock

put the **tiny** folder in the arduino's hardware folder as follows:




## step 4

open the arduino 1.0.1 and the boards list now you should see attiny as follows

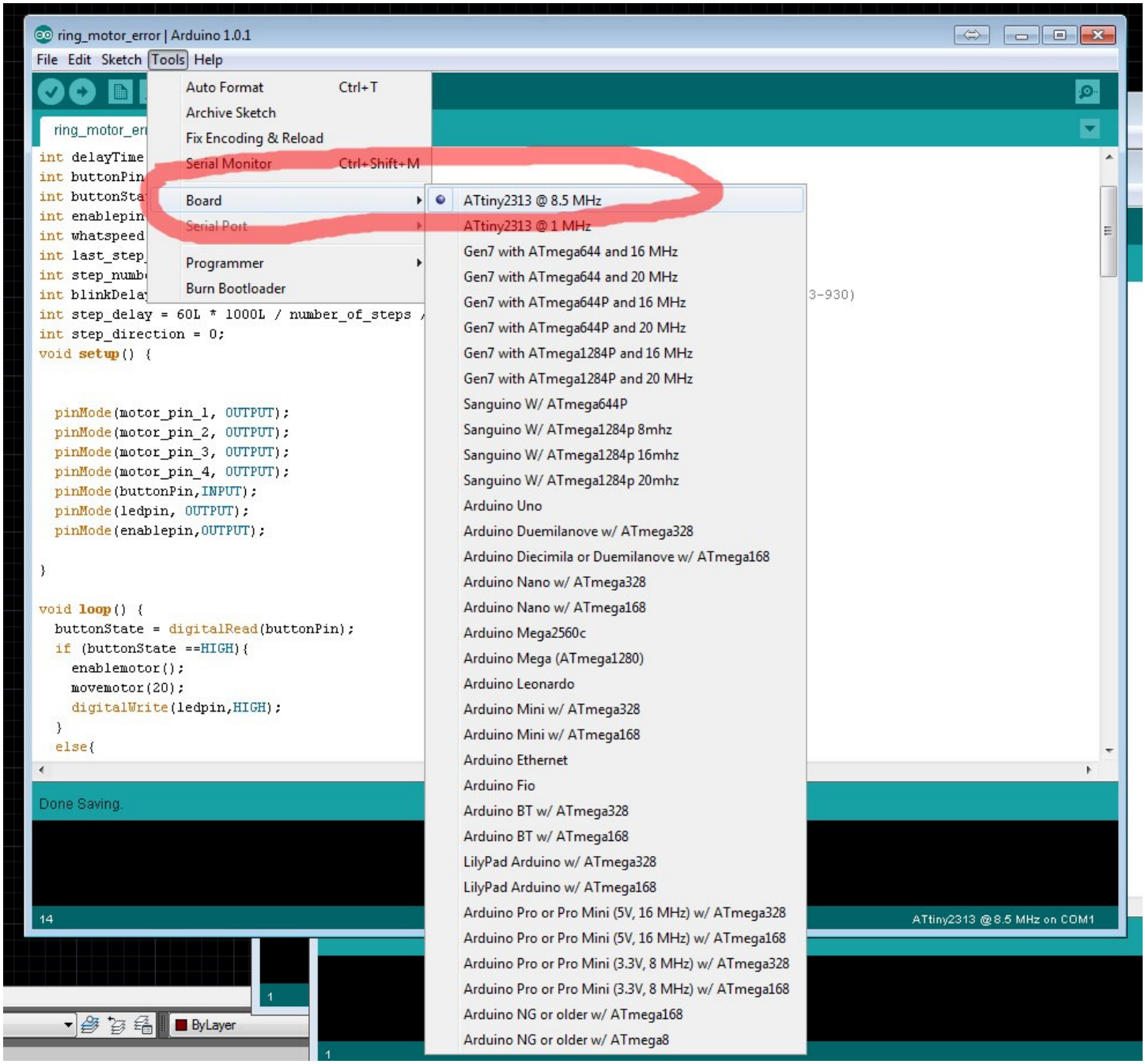


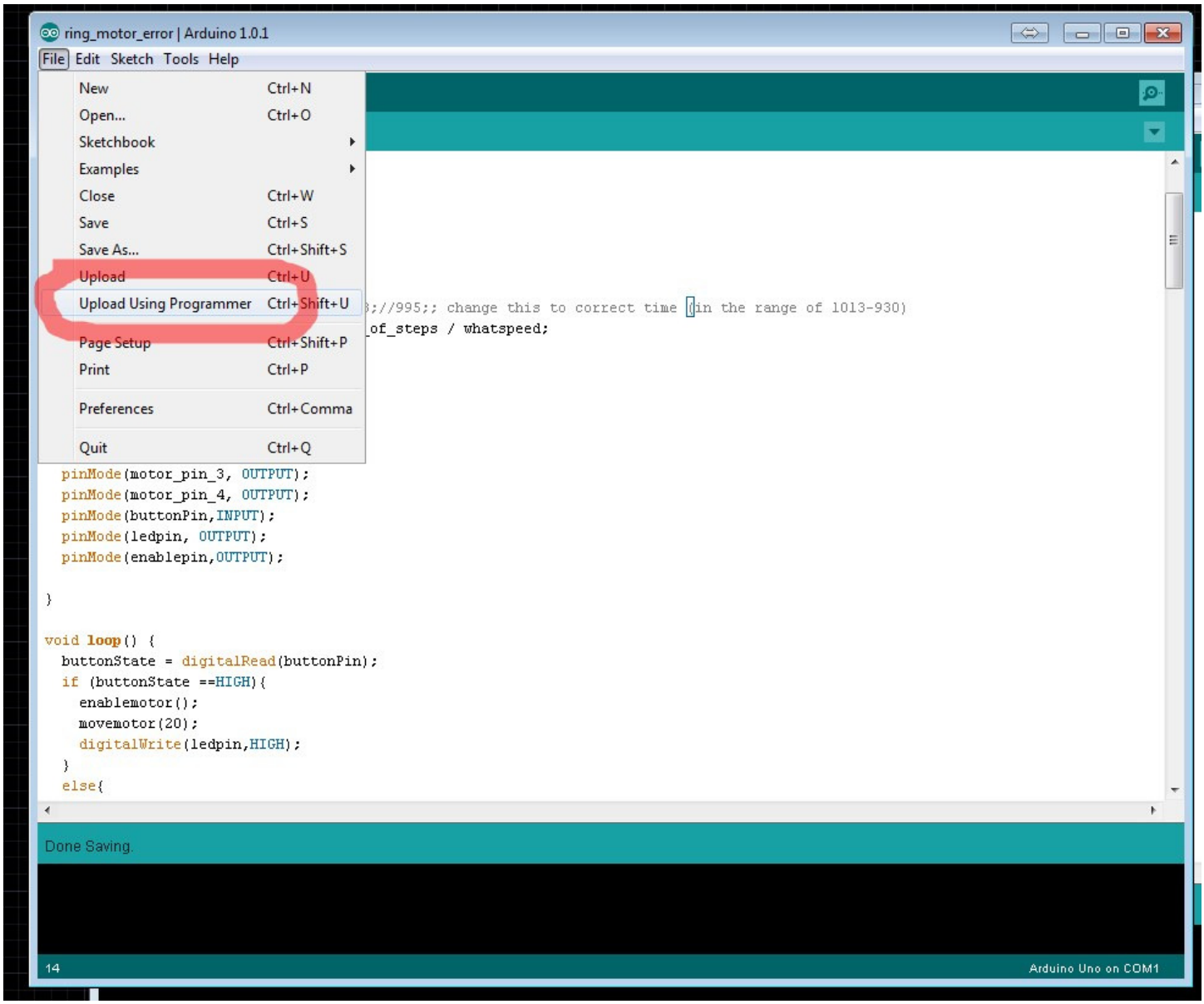

if not check the tiny folder is in the right location

## step 7

open the Arduino sketch

upload the sketch using the following method.. ( dont upload as usual because that will not work ) First select the board as attiny2313@ 8.5 . then upload using programmer


ring_motor_error | Arduino 1.0.1
File Edit Sketch Tools Help
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Serial Monitor Ctrl+Shift+M
Board
Serial Port
Programmer
Burn Bootloader
ATtiny2313 @ 8.5 MHz
ATtiny2313 @ 1 MHz
Gen7 with ATmega644 and 16 MHz
Gen7 with ATmega644 and 20 MHz
Gen7 with ATmega644P and 16 MHz
Gen7 with ATmega644P and 20 MHz
Gen7 with ATmega1284P and 16 MHz
Gen7 with ATmega1284P and 20 MHz
Sanguino W/ ATmega644P
Sanguino W/ ATmega1284p 8mhz
Sanguino W/ ATmega1284p 16mhz
Sanguino W/ ATmega1284p 20mhz
Arduino Uno
Arduino Duemilanove w/ ATmega328
Arduino Diecimila or Duemilanove w/ ATmega168
Arduino Nano w/ ATmega328
Arduino Nano w/ ATmega168
Arduino Mega2560c
Arduino Mega (ATmega1280)
Arduino Leonardo
Arduino Mini w/ ATmega328
Arduino Mini w/ ATmega168
Arduino Ethernet
Arduino Fio
Arduino BT w/ ATmega328
Arduino BT w/ ATmega168
LilyPad Arduino w/ ATmega328
LilyPad Arduino w/ ATmega168
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega328
Arduino Pro or Pro Mini (5V, 16 MHz) w/ ATmega168
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega328
Arduino Pro or Pro Mini (3.3V, 8 MHz) w/ ATmega168
Arduino NG or older w/ ATmega168
Arduino NG or older w/ ATmega8
int delayTime
int buttonPin
int buttonSta
int enablepin
int whatspeed
int last_step
int step_numb
int blinkDela
int step_delay = 60L * 1000L / number_of_steps
int step_direction = 0;
void setup() {
pinMode(motor_pin_1, OUTPUT);
pinMode(motor_pin_2, OUTPUT);
pinMode(motor_pin_3, OUTPUT);
pinMode(motor_pin_4, OUTPUT);
pinMode(buttonPin,INPUT);
pinMode(ledpin, OUTPUT);
pinMode(enablepin,OUTPUT);
}
void loop() {
buttonState = digitalRead(buttonPin);
if (buttonState ==HIGH){
enablemotor();
movemotor(20);
digitalWrite(ledpin,HIGH);
}
else{
3-930)
Done Saving.
14
ATtiny2313 @ 8.5 MHz on COM1
ByLayer

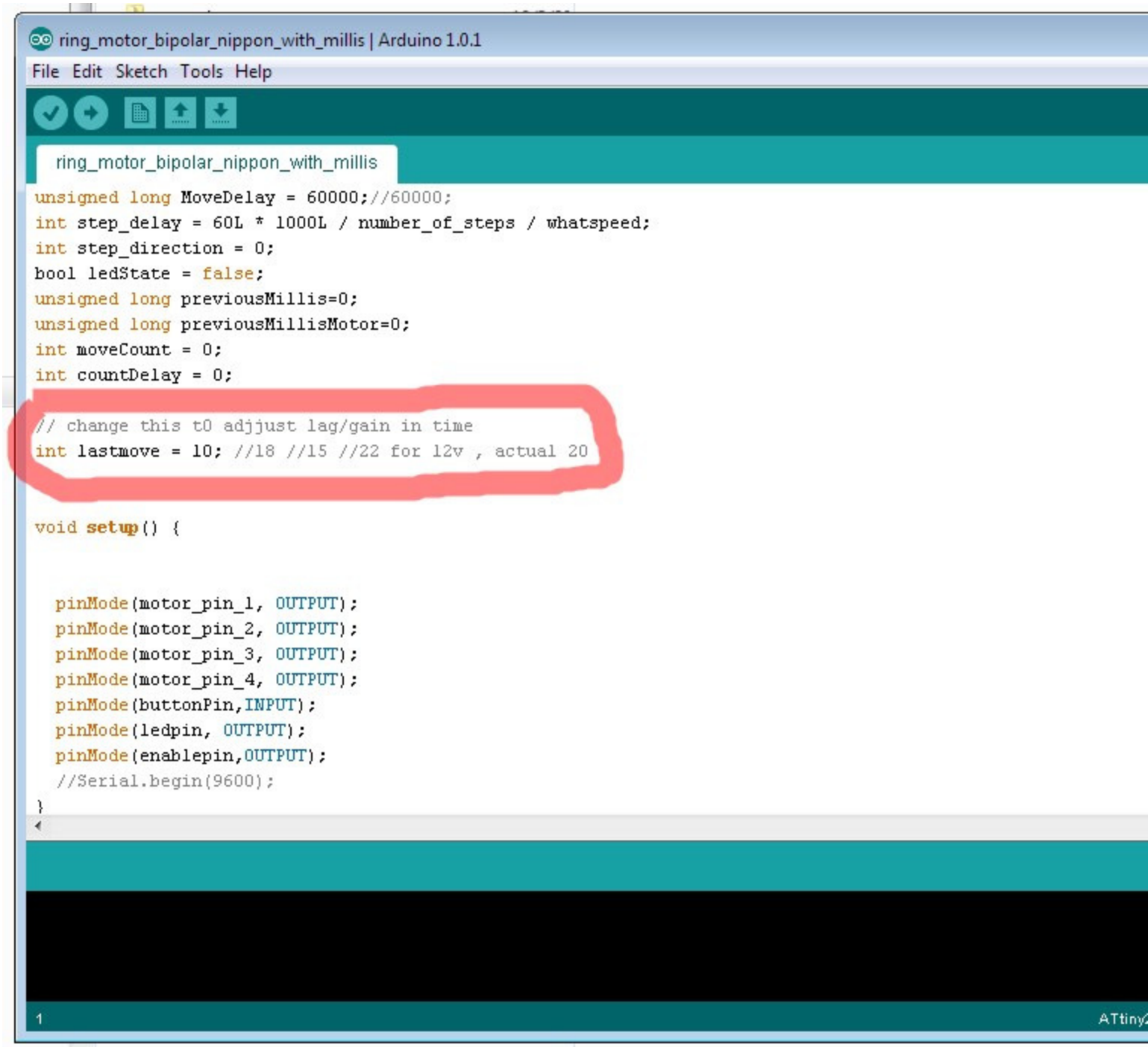



should be done ..

### step 8

If the clock is gaining or loosing time adjust the timing by changing the value for the last move

A value of 5 works well but it can vary depending on the power supply used.




That's it . the clock should be working fine now

To adjust the time just press the button on the left side and wait patiently for the needles to reach the correct position.

The clock only moves to change minutes and hours. The rest of the time it just blinks quietly to indicate seconds.