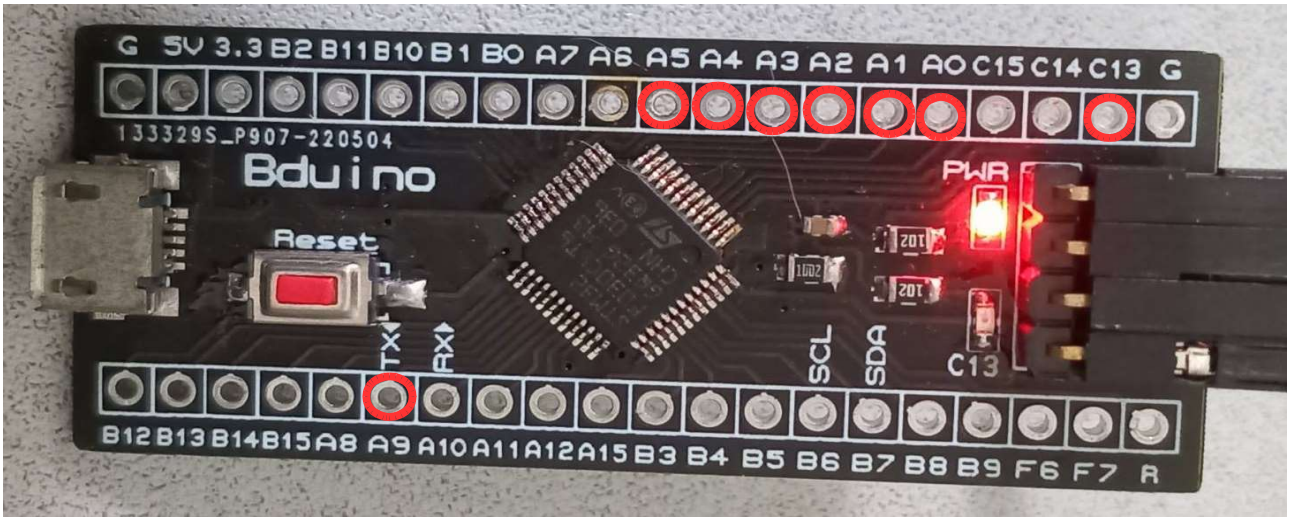
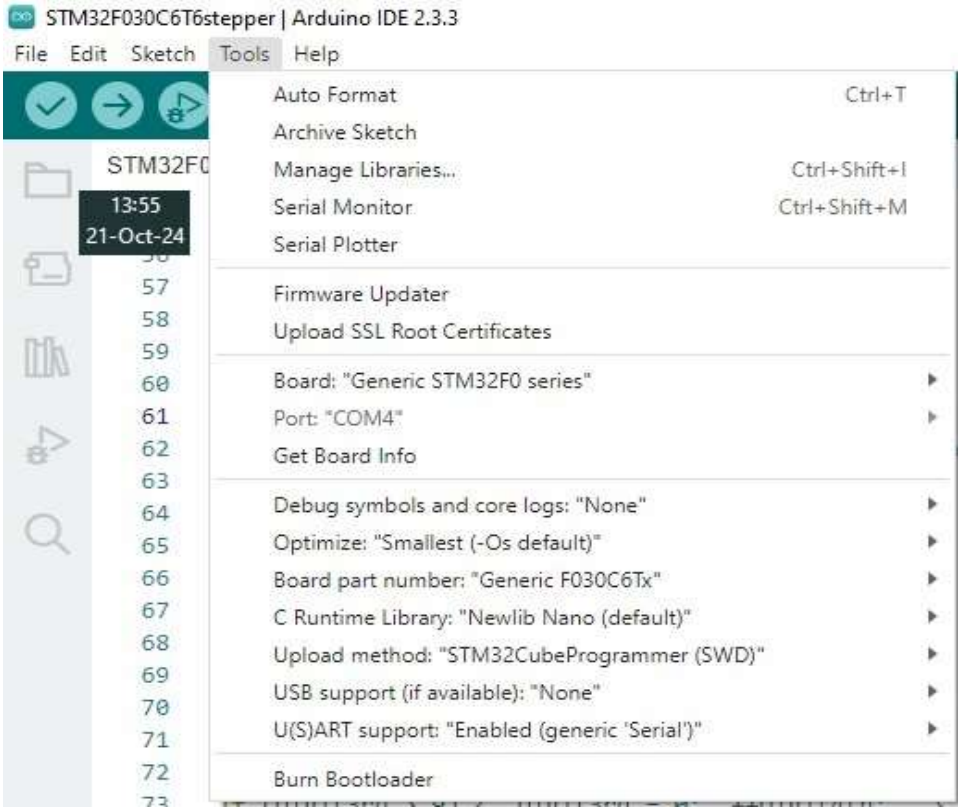
// **STM32F030C6T6**stepper.ino



/* Stepper Motor Control - one step at a time, drives a unipolar or bipolar stepper motor.
 The motor is attached to digital pins x x x x
 The motor will step one step at a time, very slowly. You can use this to
 test that you've got the four wires of your stepper wired to the correct
 pins. If wired correctly, all steps should be in the same direction.
 Use this also to count the number of steps per revolution of your motor,
 if you don't know it. Then plug that number into the oneRevolution example */

```
#include <Arduino.h>
#include <Stepper.h>

const int stepsPerRevolution = 200;  // change this to fit the number of steps per revolution for your
motor

// System clock selection is performed on startup,
// however the internal RC 8 MHz oscillator is selected as default CPU clock on reset.
// PA13 SWDIO, PA14 SWCLK, PC14/PC15 OSC,
// USART on pins PA14/PA15 or [ PA9 = USART1_TX ]    /PA10.
// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution, PA8, PA7, PA12, PA11);

int stepCount = 0;         // number of steps the motor has taken

static uint8_t Digit5th = 0; // Defines the 5th digit weight
static uint8_t Digit4th = 0; // Defines the 4th digit weight
static uint8_t Digit3rd = 0; // Defines the 3rd digit weight
static uint8_t Digit2nd = 0; // Defines the 2nd digit weight
static uint8_t tempC = 0; // Defines the last digit weight

void setup() {
Serial1.begin(2400);      // initialize the serial port:
```

```
pinMode(PA0, OUTPUT);     pinMode(PA1, OUTPUT);   pinMode(PA2, OUTPUT);
pinMode(PA3, OUTPUT);
pinMode(PA4, OUTPUT);     pinMode(PA5, OUTPUT);
//pinMode(PC0, OUTPUT);     pinMode(PC1, OUTPUT);   pinMode(PC2, OUTPUT);
pinMode(PC3, OUTPUT);
//pinMode(PC4, OUTPUT);
   //  pinMode(PC5, OUTPUT);
pinMode(PC13, OUTPUT);
Serial1.println(".....");
Serial1.print("Baud Rate 2400 : ");
Serial1.println("STM32F030C6T6 with Arduino");
Serial1.println("21 October 2024");
Serial1.println("STM32F030C6T6stepper.inO");
Serial1.println(".....");

digitalWrite(PC13, HIGH);  delay(400);  digitalWrite(PC13, LOW);   delay(400);
}

void loop() {
  //digitalWrite(PC0, !digitalRead(PC0));  delay(50);
  //digitalWrite(PC1, !digitalRead(PC1));  delay(50);
  //digitalWrite(PC2, !digitalRead(PC2));  delay(50);
  //digitalWrite(PC3, !digitalRead(PC3));  delay(50);
  // digitalWrite(PC4, !digitalRead(PC4));  delay(50);
  // digitalWrite(PC5, !digitalRead(PC5));  delay(50);
  digitalWrite(PC13, !digitalRead(PC13));  delay(50);
  digitalWrite(PA0, !digitalRead(PA0));  delay(50);
  digitalWrite(PA1, !digitalRead(PA1));  delay(50);
  digitalWrite(PA2, !digitalRead(PA2));  delay(50);
  digitalWrite(PA3, !digitalRead(PA3));  delay(50);
  digitalWrite(PA4, !digitalRead(PA4));  delay(50);
  digitalWrite(PA5, !digitalRead(PA5));  delay(50);

  myStepper.step(1);        // step one step:
  Serial1.print("steps:");  Serial1.println(stepCount);
  stepCount++;  delay(50);
   for (int i=0; i<30; i++) {
 Serial1.print(Digit5th, DEC);
 Serial1.print(Digit4th, DEC);
 Serial1.print(Digit3rd, DEC);
 Serial1.print(Digit2nd, DEC);
 Serial1.println(tempC, DEC);
 ++tempC; // increment last digit weight by one and returns the new value of x
if (tempC > 9) {  tempC = 0;  ++Digit2nd;   }            //last digit statement(s)
if (Digit2nd > 9) {  Digit2nd = 0;  ++Digit3rd;   }      //2nd digitstatement(s)
if (Digit3rd > 9) {  Digit3rd = 0;  ++Digit4th;   }      //3rd digitstatement(s)
if (Digit4th > 9) {  Digit4th = 0;  ++Digit5th;   }      //4th digitstatement(s)
if (Digit5th > 6) {  Digit5th = 0;   }                 //5th digits max = 60 tonne
 }}
```




```
Sketch uses 15380 bytes (46%) of program storage space. Maximum is 32768 bytes.
Global variables use 1268 bytes (30%) of dynamic memory, leaving 2828 bytes for local variables. Maximum is 4096 bytes.
"C:\Users\J_SOM\AppData\Local\Arduino15\packages\STMicroelectronics\tools\STM32Tools\2.2.3/win/busybox.exe" sh "C:\Users
Selected interface: swd
        -----------------------------------------------------------------
        |||||||||STM32CubeProgrammer v2.17.0
        -----------------------------------------------------------------

ST-LINK SN  : 48FF6A064885525510260287
ST-LINK FW  : V2J45S7
Board       : --
Voltage     : 3.23V
SWD freq    : 4000 KHz
Connect mode: Under Reset
Reset mode  : Hardware reset
Device ID   : 0x444
Revision ID : Rev 1.0
Device name : STM32F03x
Flash size  : 32 KBytes
Device type : MCU
Device CPU  : Cortex-M0
BL Version  : 0x10
```