Automatic Coffee Mixer with Arduino

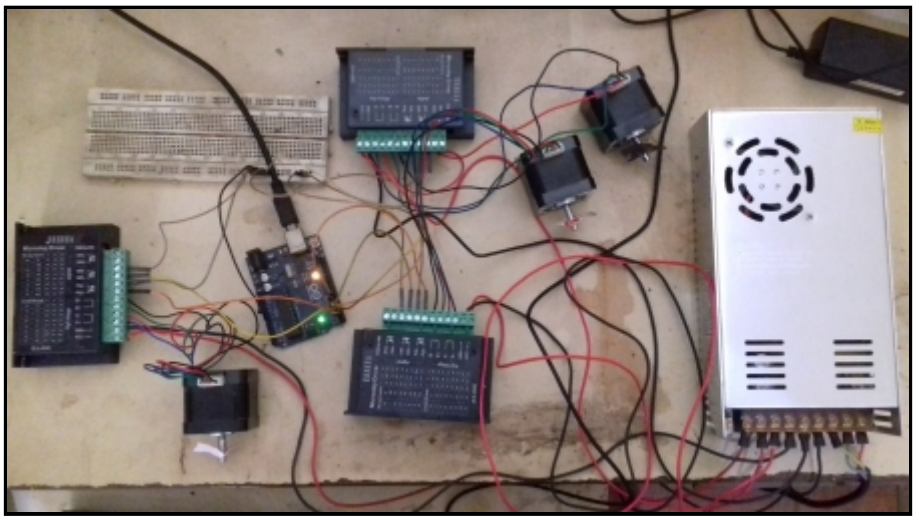
Although the author and publisher have made every effort to ensure that the information in this writing was correct at press time, the author and publisher do not assume and hereby disclaim any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from negligence, accident, or any other cause.

paid link

In this short writing, we will talk about an automatic coffee mixer that enables people to mix a certain amount of coffee and sugar in a high precision. As always, you can add more functionality to the device later (add more ingredients such as cream, etc). This small project has not yet finished but has shown good results after some trials.

Components:

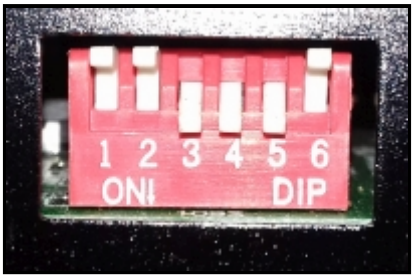
-Arduino Uno
(1) -Stepper Motor NEMA17 (3)
-TB6600 Stepper Driver (3)
-AC To DC Power Supply (220/110V to 12V/30A) (1)
-300mm Lead Screw (1)
-Breadboard (1)
-Wire



Note:

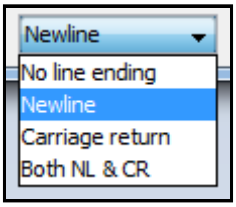
- If you use a dual voltage power supply (220/110V input), ensure that the selection switch is set properly.

- Do not forget to set the driver's switches. The right setting depends on the stepper motor that you use (the peak current).



- The wiring between the stepper driver and the stepper motor may be different from the circuit presented here.

- Set the Serial Monitor so that it accepts newline (' ').



Here is the code for Arduino:

```
int driverDIR1 = 2;
int driverPUL1 = 3;
int driverDIR2 = 4;
int driverPUL2 = 5;
int driverDIR3 = 6;
int driverPUL3 = 7;
```

```
int pd = 500;   //pulse delay period

String inputString = '';
boolean inputComplete = false;
boolean inputValid = false;

int numCoffee = 0;
int numSugar = 0;

void setup() {
  Serial.begin(9600);

  pinMode(driverDIR1,OUTPUT);
  pinMode(driverPUL1,OUTPUT);
  pinMode(driverDIR2,OUTPUT);
  pinMode(driverPUL2,OUTPUT);
  pinMode(driverDIR3,OUTPUT);
  pinMode(driverPUL3,OUTPUT);
}

void loop() {
  serialEvent();

  if (inputComplete) {
    Serial.println('inputComplete');
    checkInput();
    if (inputValid){
      Serial.println('inputValid');
      runDevice();
    }
    inputString = '';
    inputComplete = false;
    inputValid = false;
    numCoffee = 0;
    numSugar = 0;
  }

  delay(1000);
}

void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();
    if (inChar == ' ') {      //or inChar == 10
      inputComplete = true;
    }else{
      inputString += inChar;
    }
  }
}

void checkInput() {
  if(inputString.length() < 2){
    return;
  }

  numCoffee = (inputString.substring(0,1)).toInt();
  if ((numCoffee >= 1) %26;%26; (numCoffee <= 5)){
    numSugar = (inputString.substring(1,2)).toInt();
    if ((numSugar >= 1) %26;%26; (numSugar <= 5)){
      inputValid = true;
    }
  }
}

void runDevice(){
  Serial.println('runDevice');
  runTrack(10);     //go to the coffee contain
  delay(1000);
  dropCoffee(numCoffee);
  delay(1000);
```

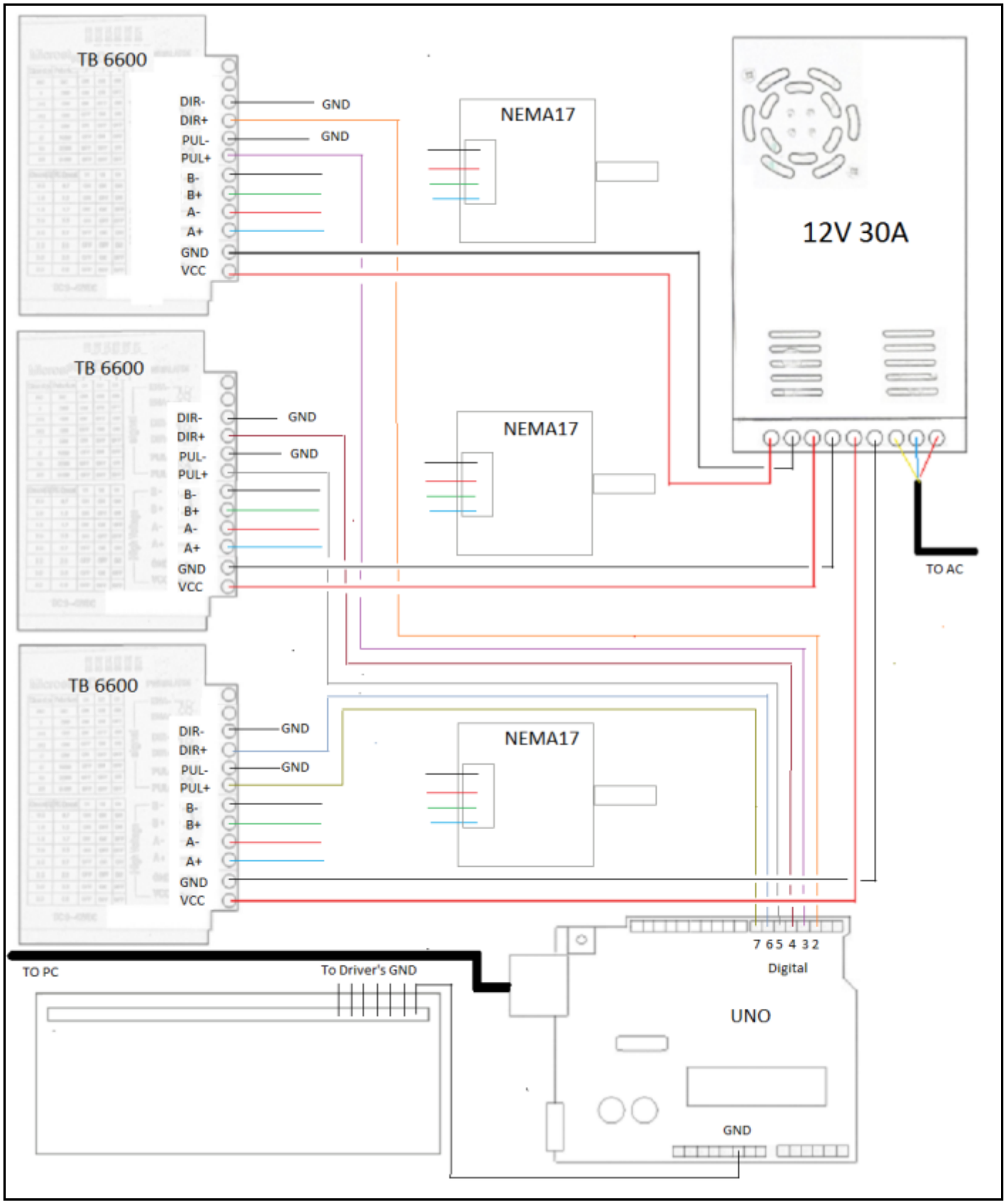
```
  runTrack(5);     //go to the sugar container
  delay(1000);
  dropSugar(numSugar);
  delay(1000);
  runTrack(10);     //go to the finish line
  Serial.println('Finished');
}

//stepper1
void runTrack(int number){
  Serial.println('runTrack');
  number = number * 500;
  for (int i = 0; i < number; i++) {
    //Serial.println(i);
    digitalWrite(driverDIR1, HIGH);
    digitalWrite(driverPUL1, HIGH);
    delayMicroseconds(pd);
    digitalWrite(driverPUL1, LOW);
    delayMicroseconds(pd);
  }
}

//stepper2
void dropCoffee(int number){
  Serial.println('dropCoffee');
  number = number * 500;
  for (int i = 0; i < number; i++) {
    //Serial.println(i);
    digitalWrite(driverDIR2, HIGH);
    digitalWrite(driverPUL2, HIGH);
    delayMicroseconds(pd);
    digitalWrite(driverPUL2, LOW);
    delayMicroseconds(pd);
  }
}

//stepper3
void dropSugar(int number){
  Serial.println('dropSugar');
  number = number * 500;
  for (int i = 0; i < number; i++) {
    //Serial.println(i);
    digitalWrite(driverDIR3, HIGH);
    digitalWrite(driverPUL3, HIGH);
    delayMicroseconds(pd);
    digitalWrite(driverPUL3, LOW);
    delayMicroseconds(pd);
  }
}
```

Here is the Circuit:



Explanation

The variables *driverDIR1* and *driverPUL1* drive stepper motor 1. The motor is responsible to rotate the lead screw. This in turn will move a small platform for the coffee cup. The variables *driverDIR2* and *driverPUL2* drives stepper motor 2. The motor rotates a small wheel inside a container (for coffee). The variables *driverDIR3* and *driverPUL3* drives stepper motor 3. The motor rotates a small wheel inside a container (for sugar).

The variables *numCoffee* and *numSugar* determine the amount of coffee and sugar that will be poured into the glass. The variables hold the number of rotation of the wheels. Put it simple, user input will be stored here.

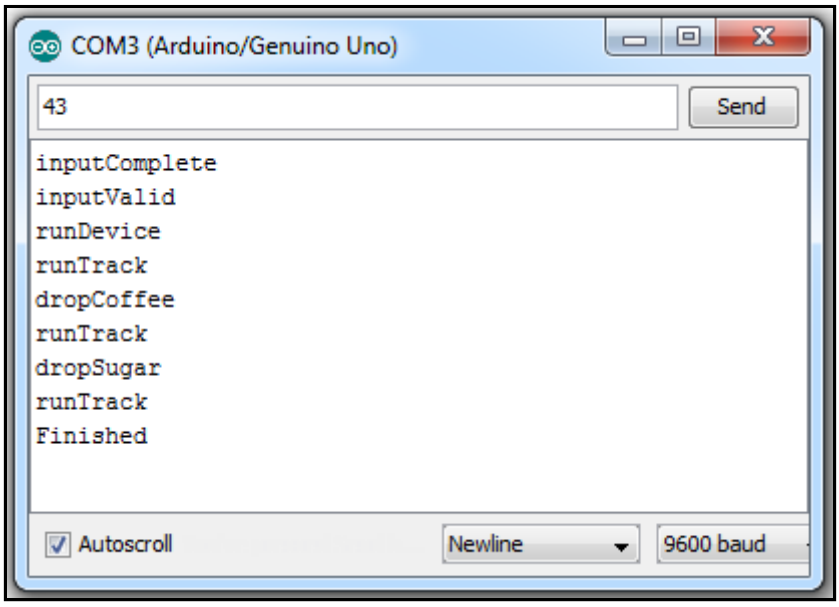
```
void serialEvent() {
  while (Serial.available()) {
    char inChar = (char)Serial.read();
    if (inChar == ' ') {
      inputComplete = true;
    }else{
      inputString += inChar;
    }
  }
}
```

User input is sent through the Arduino IDE (Serial Monitor). The variable *inputString* holds user input. Anytime you click the 'send' button, input checking is performed.

```
void checkInput() {
  if(inputString.length() < 2){
    return;
  }

  numCoffee = (inputString.substring(0,1)).toInt();
  if ((numCoffee >= 1) %26;%26; (numCoffee <= 5)){
    numSugar = (inputString.substring(1,2)).toInt();
    if ((numSugar >= 1) %26;%26; (numSugar <= 5)){
      inputValid = true;
    }
  }
}
```

Based on the above lines of code, valid input must consists of two integer values. The first value deals with the amount of coffee to mix and the second value determines the amount of sugar that you want to add to your coffee. Have a look at the IDE below:




```
void runDevice(){
  Serial.println('runDevice');
  runTrack(10);     //go to the coffee contain
  delay(1000);
  dropCoffee(numCoffee);
  delay(1000);
  runTrack(5);     //go to the sugar container
  delay(1000);
  dropSugar(numSugar);
```

```
  delay(1000);
  runTrack(10);     //go to the finish line
  Serial.println('Finished');
}
```

If user input is valid, the process begins. The platform for the coffee cup moves to the coffee container (*runTrack(10)*). The wheel inside the coffee container rotates (*dropCoffee(numCoffee)*). The platform is then moved to the sugar container (*runTrack(5)*). The wheel inside the sugar container rotates (*dropSugar(numSugar)*). Finally, the coffee cup goes to the finish line (*runTrack(10)*).

```
void runTrack(int number){
  Serial.println('runTrack');
  number = number * 500;
  for (int i = 0; i < number; i++) {
    //Serial.println(i);
    digitalWrite(driverDIR1, HIGH);
    digitalWrite(driverPUL1, HIGH);
    delayMicroseconds(pd);
    digitalWrite(driverPUL1, LOW);
    delayMicroseconds(pd);
  }
}
```

The code for all the steppers are similar. Change the number '500' in the above code as needed. Your chosen microstepping, the placement of the container, are all determine the right number to choose. Finding the right number will require some trials.