★ Differentiate between READ and WRITE pins:

→ digitalRead() works on all pins, used to read data received.
→ digitalWrite() is used to write a HIGH or a LOW value to a digital pin.
→ analogWrite() can be used to light an LED at varying brightness or drive motors.
→ analogRead() reads the value from specified analog pin.

READ

```
int b=0;
void setup()
{ Serial.begin(9600); }
void loop()
{  if (Serial.available()>0)
   { b = Serial.read();
     Serial.print("received");
     Serial.print(b,DEC); }
```

WRITE

```
void setup()
{ Serial.begin(9600); }
Void loop()
{ Serial.Write(45); }
```

★ Soldering - is used to make electrical connections. A soldering iron is used to heat the base metal of the part to be soldered & solder is then melted onto the metal, to create an alloy of the metal & to connect surface..

Program - 2

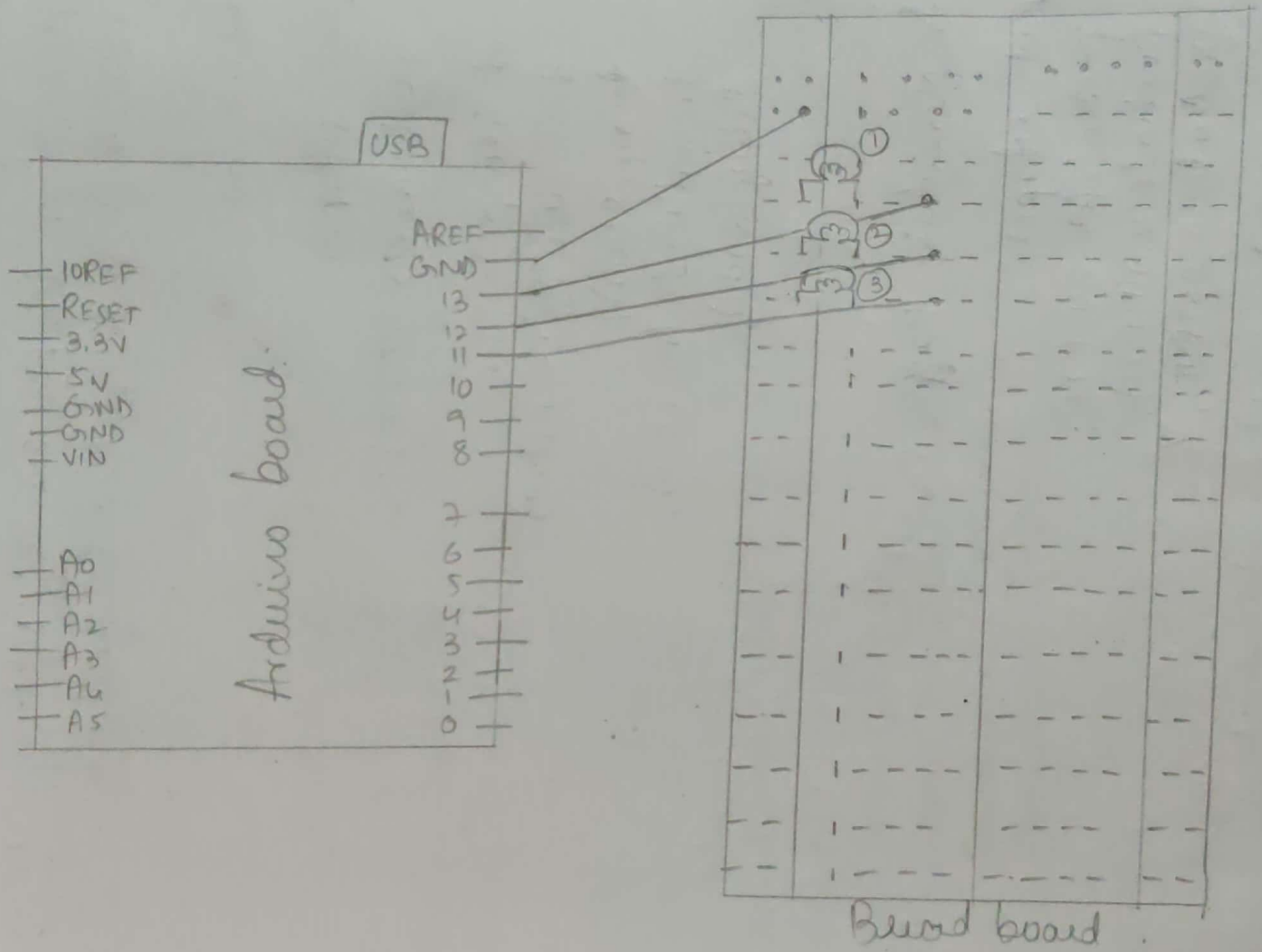
Arduino program to blink an LED and implement a traffic signal system using digitalWrite() and pinMode() functions.

a) Blink an LED -

```
void setup()
{  pinMode(13, OUTPUT); }

void loop()
{  digitalWrite(13, HIGH);
   delay(2000);
   digitalWrite(13, LOW);
   delay(2000);
}
```

b) Implementing traffic signal -

```
void setup()
{ pinMode(13, OUTPUT);
  pinMode(12, OUTPUT);
  pinMode(11, OUTPUT); }

Void loop(int x, int y, int z, int a)
{     digitalWrite(x, HIGH); delay(a);
      digitalWrite(y, LOW); digitalWrite(z, LOW); }

void loop()
{  loop1(13, 12, 11, 4000); //red
   loop1(12, 11, 13, 2000); //green
   loop1(11, 13, 12, 1000); // yellow
}
```

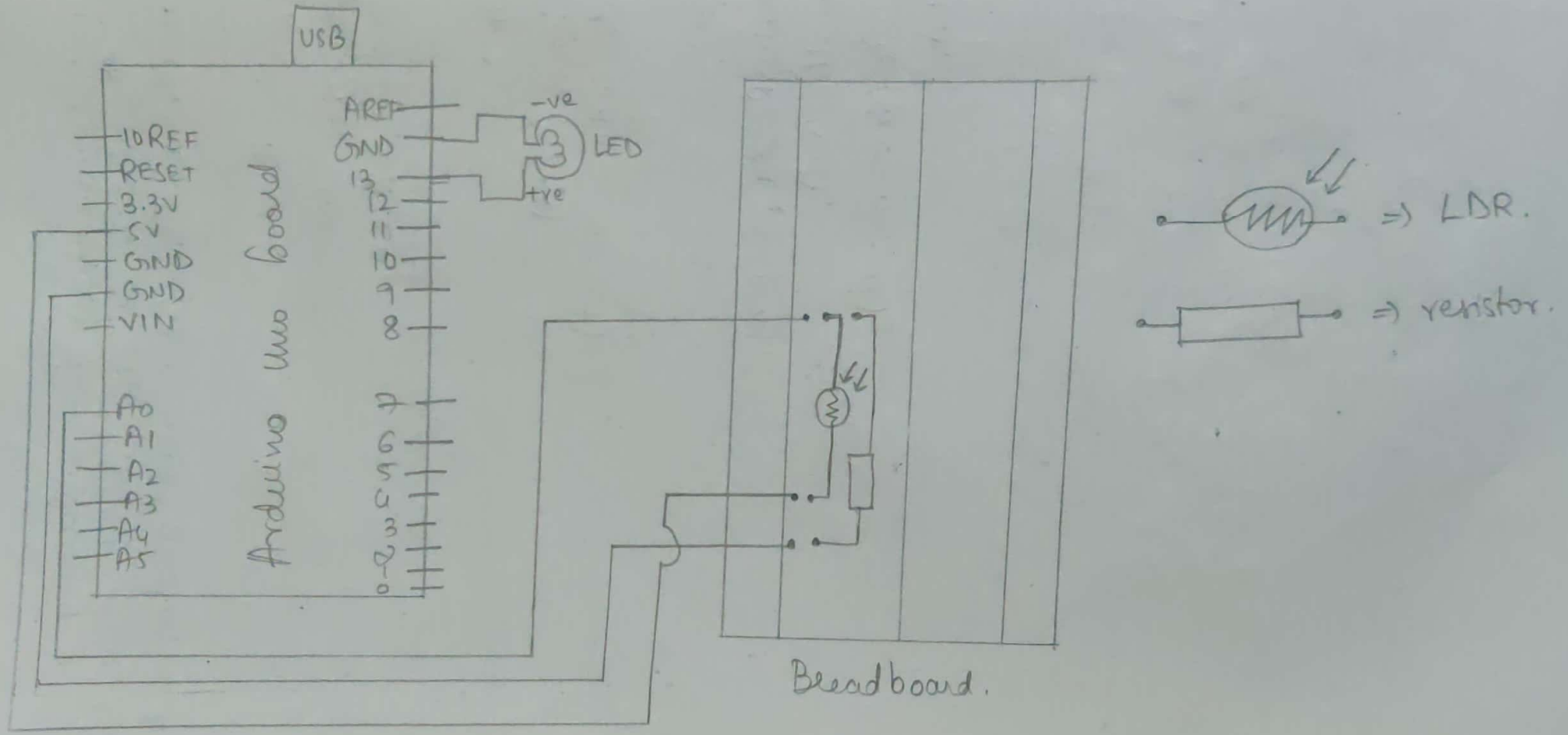


① — Red light
② — green light
③ — yellow light

Program 3

Arduino program to vary the intensity of LED based on the reading of LDR (light dependent resistor) using analogRead() and analogWrite() functions.

```
int val=0;
int postpin=0;   //Ao
int ledpin=13;   //LED

void setup()
{  Serial.begin(9600);
   pinMode(ledpin, OUTPUT); }

void loop()
{  val=analogRead(postpin);
   Serial.println(val);
   analogWrite(ledpin,val);
   delay(10);
}
```



Program 4

Arduino program to toggle LED by pressing a button and to implement a switch debounce circuit to prevent glitches in user input

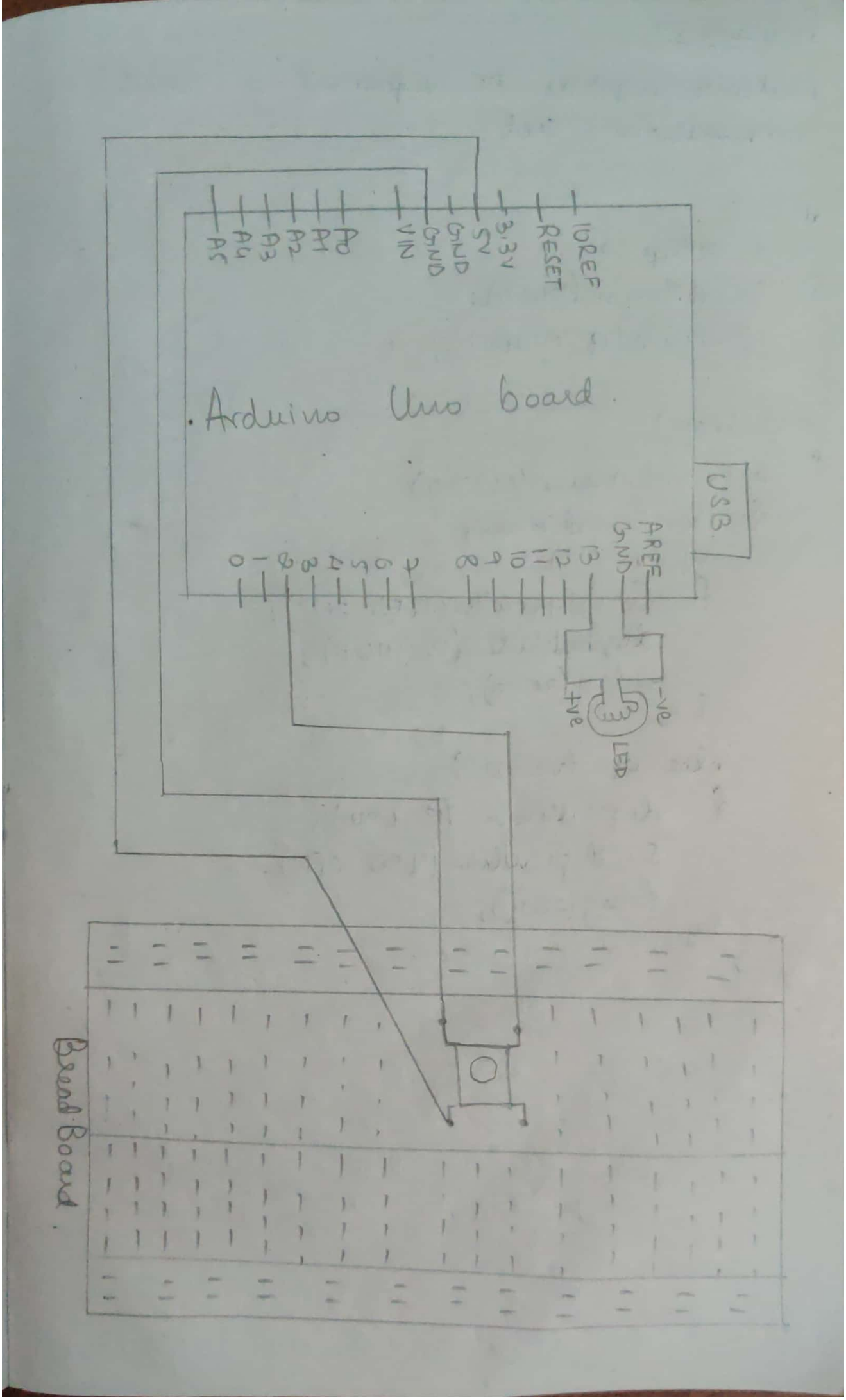
a) Button:

```
const int bp = 2;
const int lp = 13;
int buttonstate = 0;

void setup()
{ pinMode (lp, OUTPUT);  pinMode (bp, INPUT); }

void loop()
{ buttonstate = digitalRead (buttonpin);
  if (buttonstate == HIGH)
  { digitalWrite (lp, HIGH); }
  else
  { digitalWrite (lp, LOW); }
}
```

b) debounce:

```
Const int bp = 2; // buttonpin
Const int lp = 13; // ledpin
int buttonstate;
int ls = HIGH; // ledstate
int lbs = LOW; // lastbuttonstate
unsigned long lastdebouncetime = 0;
unsigned long debouncedelay = 50,

void setup()
{ pinMode (bp, INPUT);
  pinMode (lep, OUTPUT); digitalWrite (lp, ls); }
```

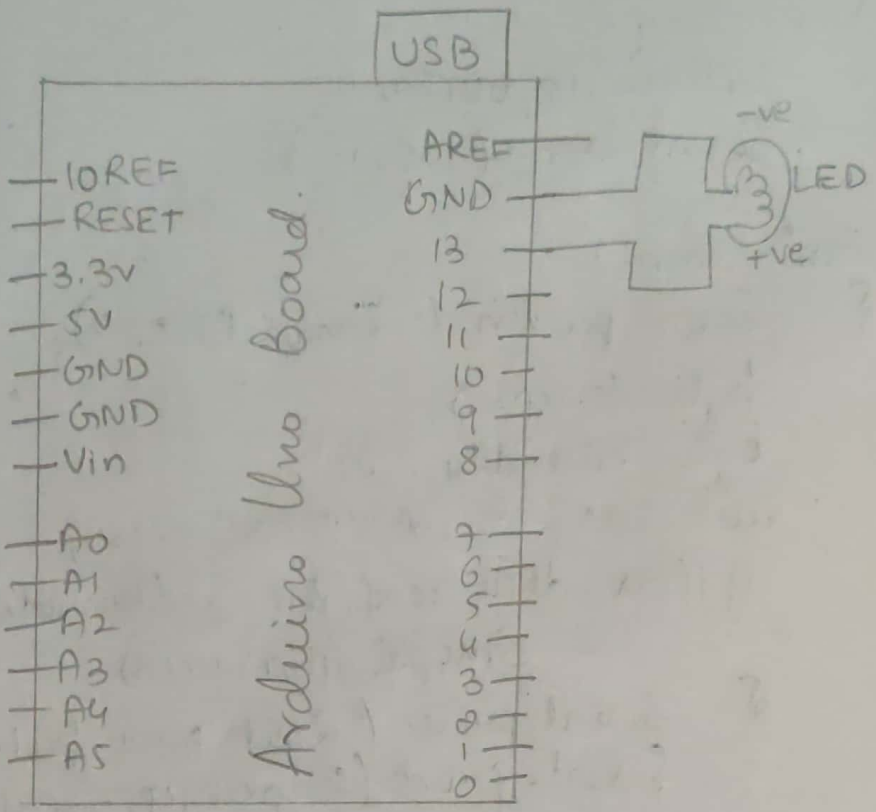
```
Void loop()
{ int reading = digitalRead (bp);
  if (reading != lbs)
  { lastdebouncetime = millis (); }
  if ((millis()- lastdebouncetime) > debouncedelay)
  { if (reading != bs)
    { bs = reading;
      if (bs == HIGH)
      { ls = ! ls; }  }  }
  digitalWrite (lp, ls);
  lbs = reading;
}
```



Program 5

Arduino program to implement a serial communication event

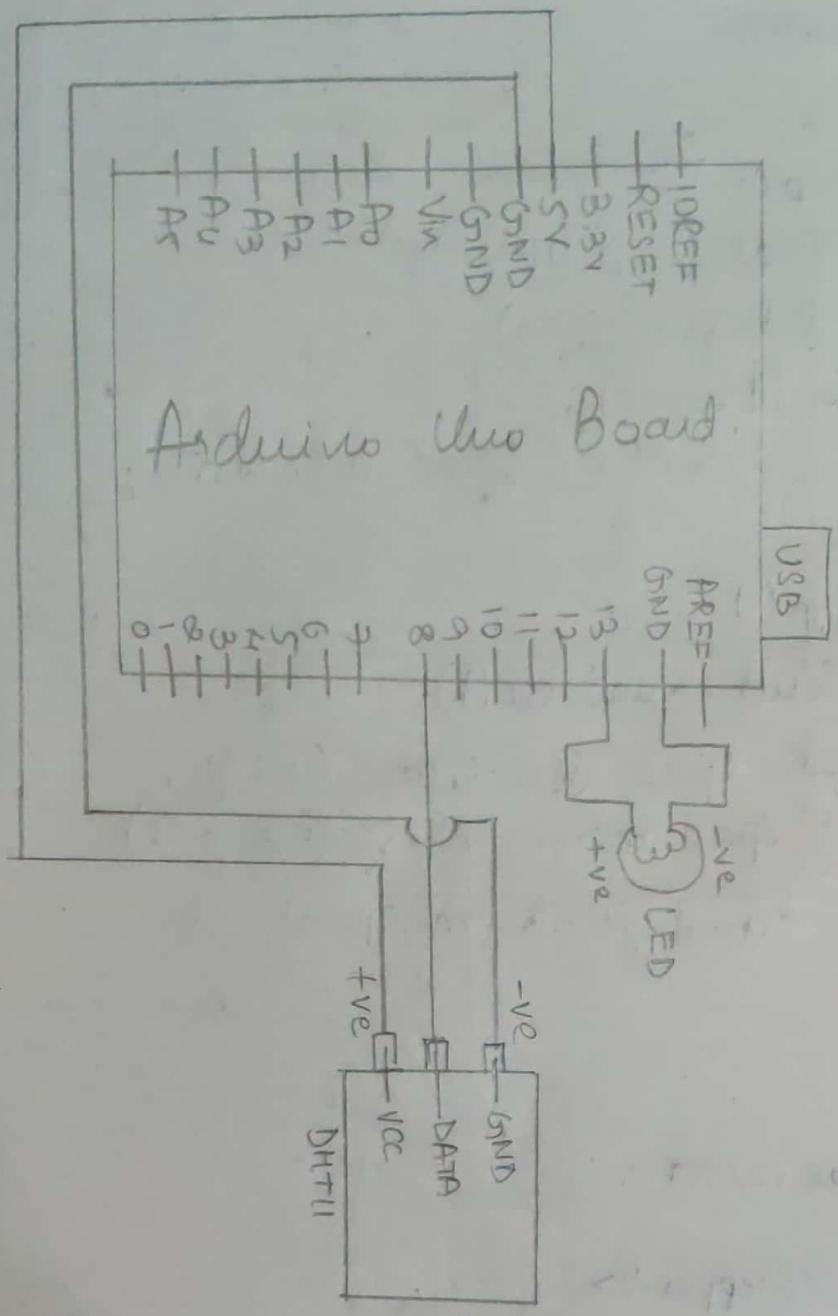
```
int v;
void setup()
{ Serial.begin(9600);
  pinMode(13, OUTPUT); }

void loop()
{ if(Serial.available()>0)
  { v = Serial.read();
    if (v=='1')
    { Serial.println("LED ON");
      digitalWrite(13, HIGH);
      delay(2000);
    }
    else if (v=='0')
    { digitalWrite(13, LOW);
      Serial.println("LED OFF");
      delay(2000);
    }
  }
}
```



Program 6

Arduino program to implement a temperature and humidity sensor and switch ON an LED if the temperature is too hot.

```
#include <SimpleDHT.h>
int pin=8;
SimpleDHT11 dht11(pin);

void setup()
{ pinMode(13,OUTPUT);
  Serial.begin(9600); }

void loop()
{ Serial.println("SimpleDHT11");
  byte temp=0;
  byte humidity=0;
  int err=SimpleDHTErrSuccess;
  if((err=dht11.read(&temp,&humidity,NULL))!=
        SimpleDHTErrSuccess)
  { Serial.print("Read DHT11 failed, err=");
    Serial.print(SimpleDHTErrCode(err));
    Serial.print(",");
    Serial.println(SimpleDHTErrDuration(err));
    delay(1000);
    return;
  }
  Serial.print("Sample OK: ");
  Serial.print((int)temp); Serial.print("*C,");
  Serial.print((int)humidity); Serial.print("H");
  if(humidity>71)
  { digitalWrite(13,HIGH); }
  else { digitalWrite(13,LOW); } delay(3000); }
```



Program 7

Arduino program to drive a DC motor and a Stepper motor.

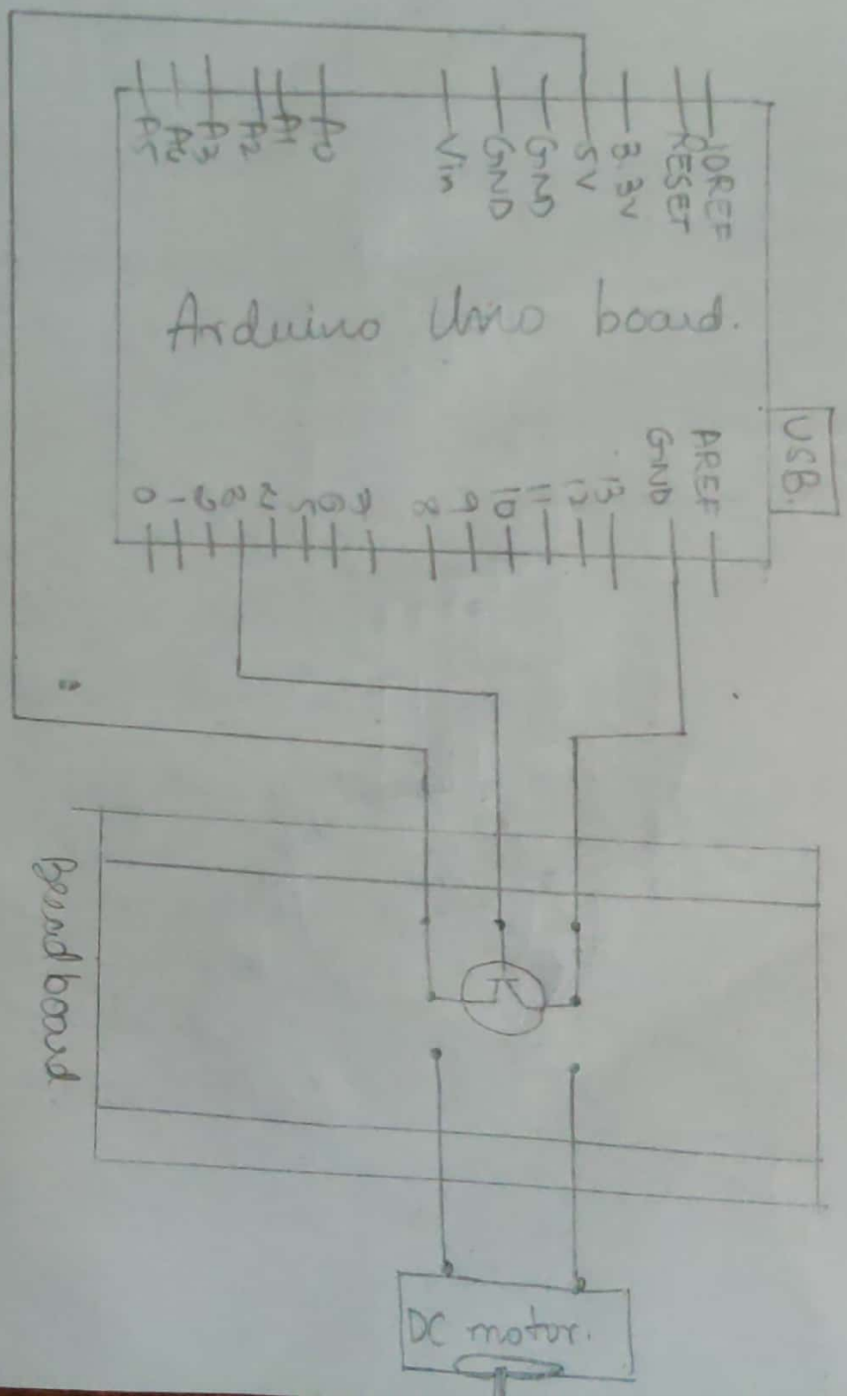
a) DC motor:

```
int m=3;
void setup()
{  pinMode (m, OUTPUT);
   Serial.begin (9600);
   while (!Serial);
   Serial.println ("Speed 0 to 255");
}
void loop()
{  if (Serial.available())
   {  int speed = Serial.parseInt();
      if(speed >=0 && speed <=255)
      {  analogWrite (m, speed); }
   }
}
```

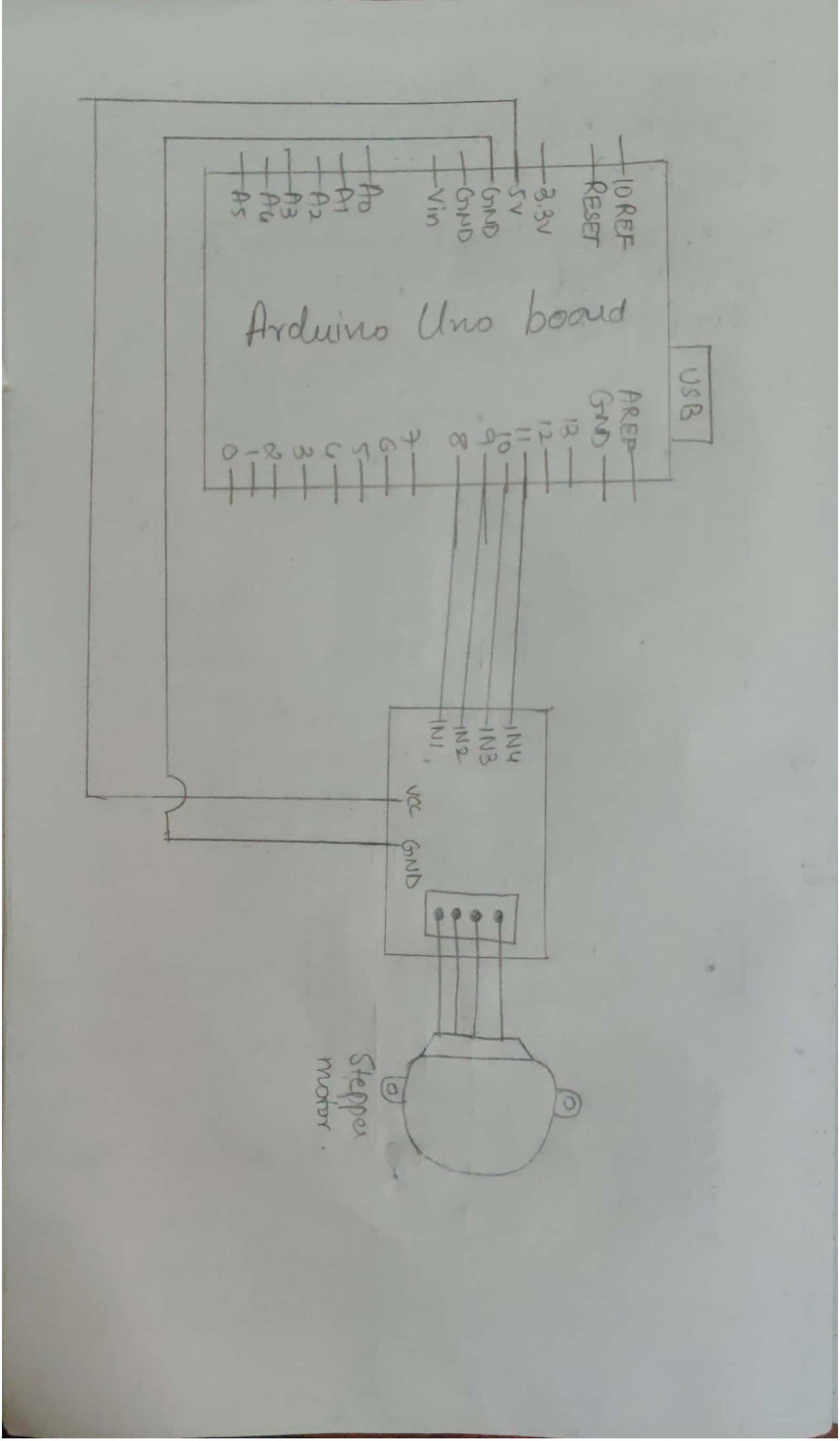
b) Stepper motor:

```
#include <Stepper.h>
const int spr=200; // steps per resolution
Stepper myStepper (spr, 8, 9, 10, 11);

void Setup()
{  myStepper.setSpeed (60);
   Serial.begin (9600); }
```

```
void loop()
{ Serial.println("clockwise");
  myStepper.step(spr);
  delay(500);
  Serial.println("counterclockwise");
  myStepper.step(-spr);
  delay(500);
}
```



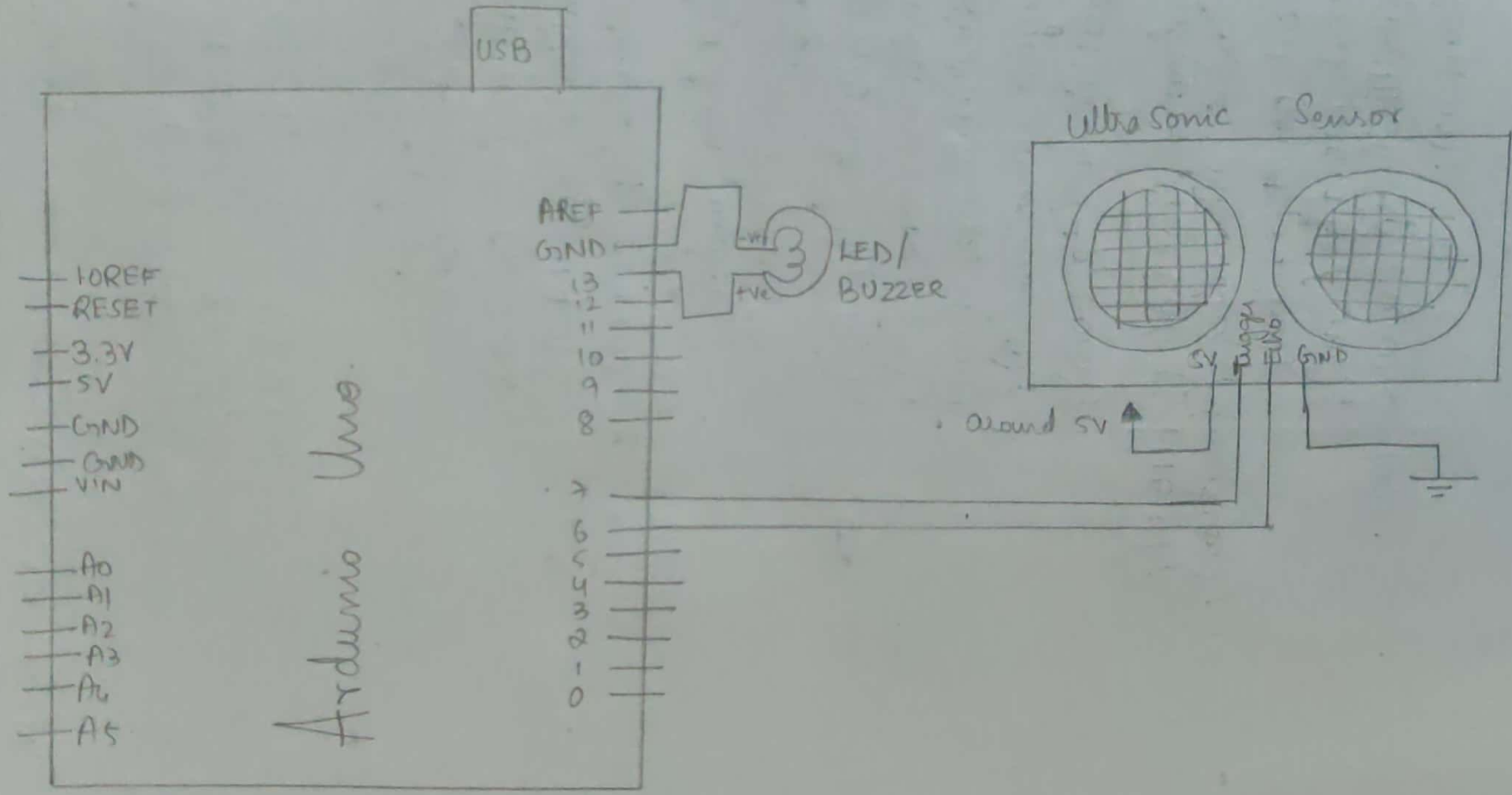
Pin diagram for DC motor implementation.



Program 8

Arduino program to implement an ultrasonic sensor to measure distance to an obstacle and "buzz" when too close to object.

```
const int tp = 7; //triggerpin
const int ep = 6; //echopin
const int lp = 13; //led pin

void setup()
{  Serial.begin(9600);
   pinMode(tp, OUTPUT);
   pinMode(ep, INPUT);
   pinMode(lp, OUTPUT); }

long duration, inches;

void loop()
{  digitalWrite(tp, LOW);
   delayMicroseconds(2);
   digitalWrite(tp, HIGH);
   delayMicroseconds(10);
   digitalWrite(tp, LOW);

   duration = pulseIn(ep, HIGH);
   inches = msToInches(duration);
   Serial.print(inches);
   Serial.println("in");
   digitalWrite(lp, inches <= 100);
   delay(100);
}

long msToInches(long ms)
{  return ms / 74 / 2; }
```



Pin diagram