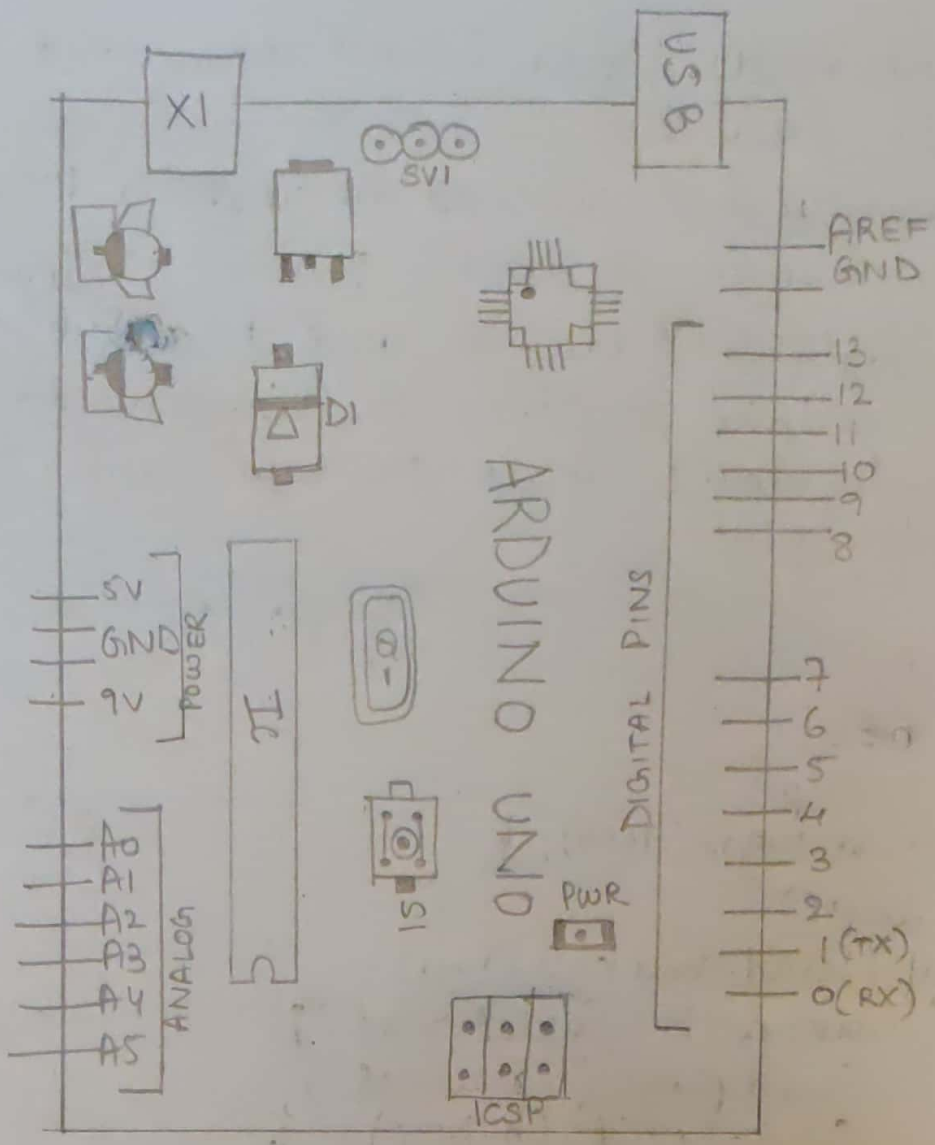
Program-1

Learn the basics of Arduino MCU boards, features and pinouts of Arduino UNO, differentiate b/w READ & WRITE pins, install and configure the Arduino IDE and basics of soldering.

→ Reset Button - S1
→ Analog pins A0 - A5
→ X1 - external power supply in - 9-12V DC
→ SV1 - Toggles external power & USB power
→ USB - used for uploading sketches to the board & for Serial communication b/w board & computer.
→ Digital pins - 13 to 1
→ Digital pins - 0-1/ serial in/out - TX/RX (these pins can't be used for digital I/O)
→ ICSP - In-circuit Serial programmer
→ AREF - reference voltage for analog inputs
→ VIN - (labelled as 9V)
→ 5V - regulated power supply
→ 3V3 or 3.3V - a 3.3 volts supply
→ GND - ground pins
→ Rx - used to recieve serial data
→ TX - used to transmit serial data.



* Basic principles of soldering:

1. The solder should have a natural glossy shine.
2. There should be clearly visible lenies
3. There should be correctly shaped fillets
4. Contact angle (θ) should be small
5. There should be no cracks or pinholes.

★ 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 & the connect surface.

Program-2

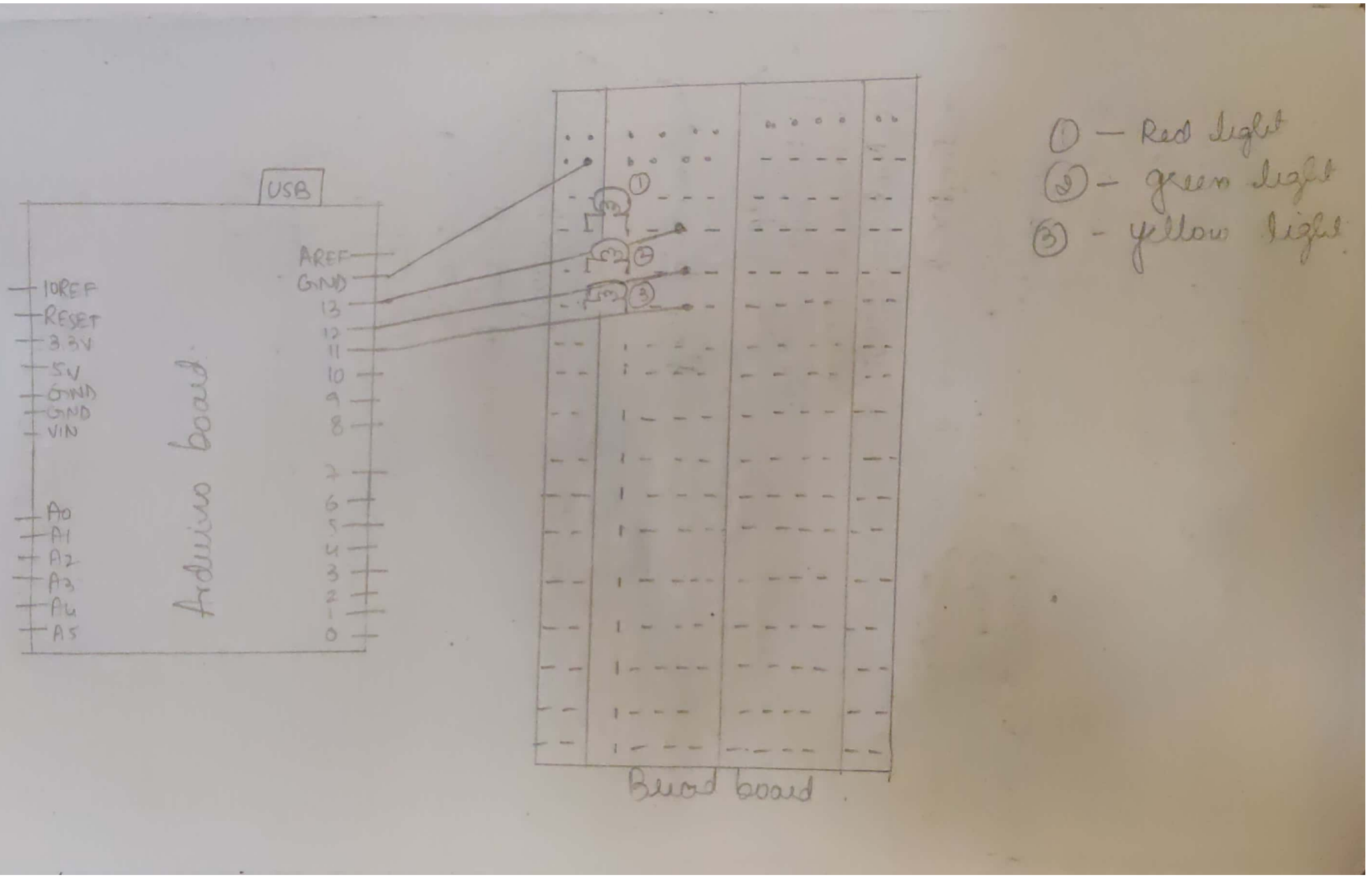
Arduino program to blink an LED and implement a traffic signal system using digitalWrite() and pinMode() functions.

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 -

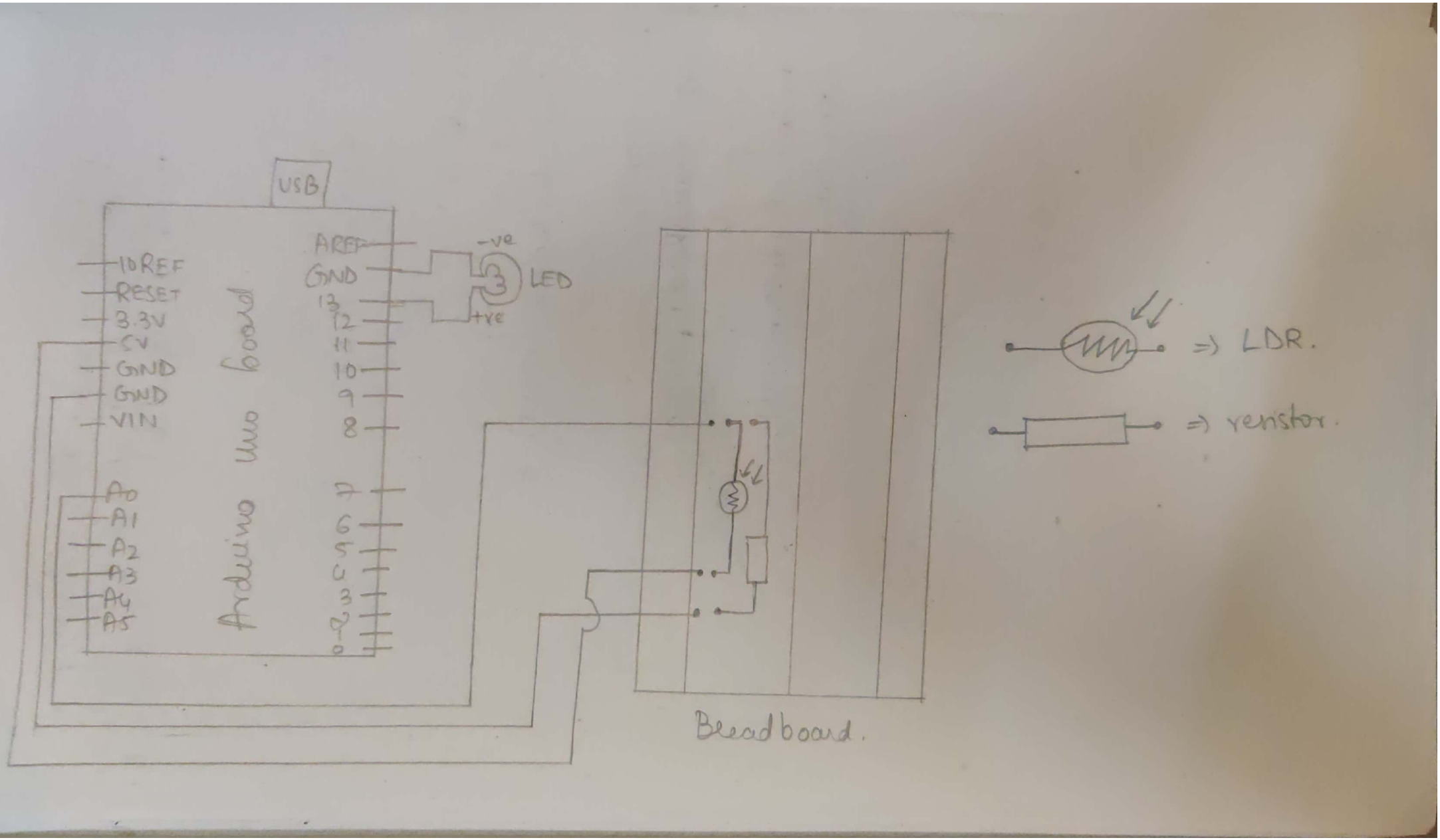
```
void setup()
{ pinMode(13, OUTPUT);
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT); }

void loop1(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
}
```



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;  //A0
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:

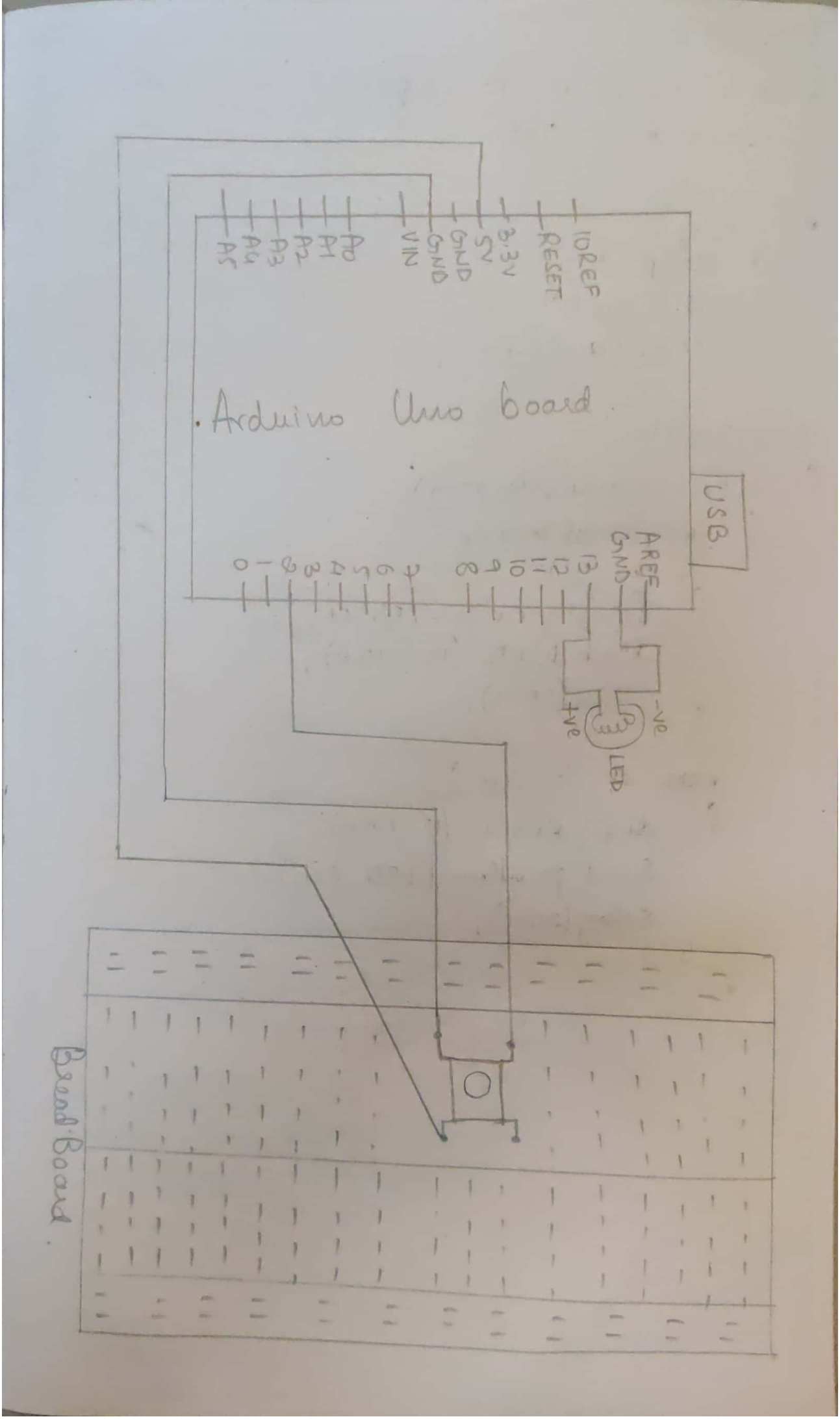
```
const int bp = 2;
const int lp = 13;
int buttonstate = 0;

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 (lp, 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;
}
```

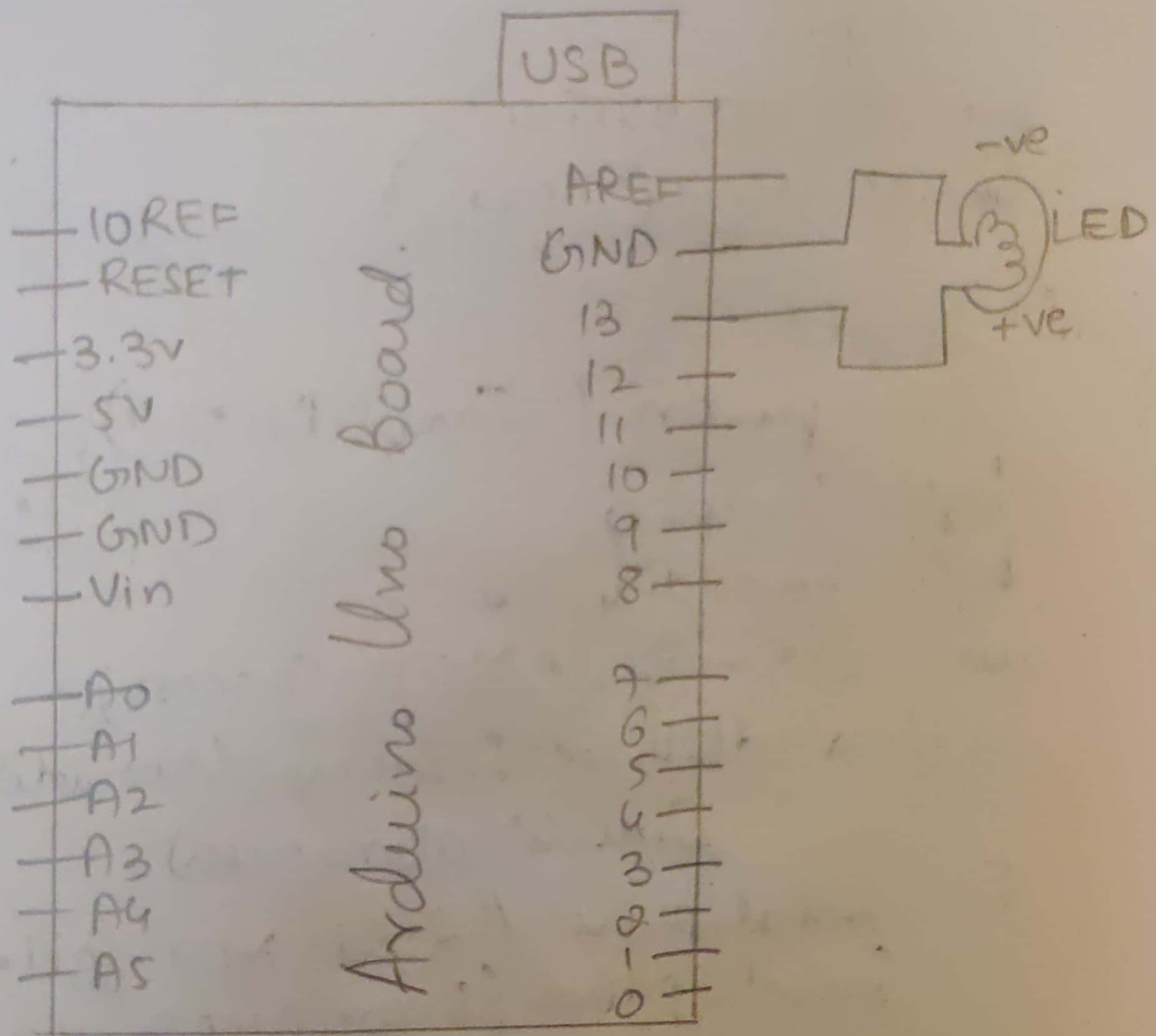
Bread Board



Program 5

Arduino program to implement a serial communication event

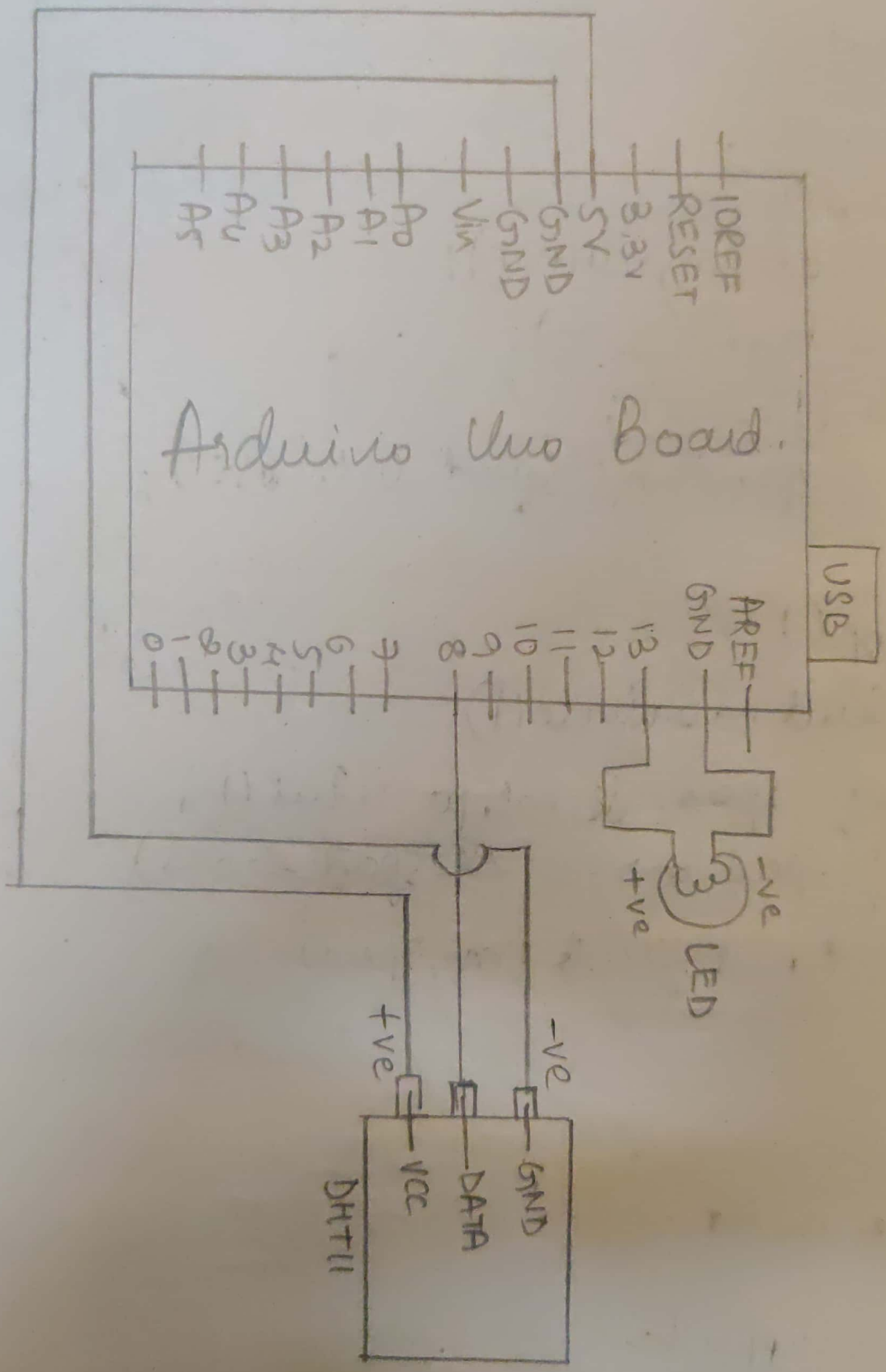
```
int v;
void setup()
{ Serial.begin(9600);
  pinMode(13, OUTPUT); }

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("SampleDHT11");
  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:

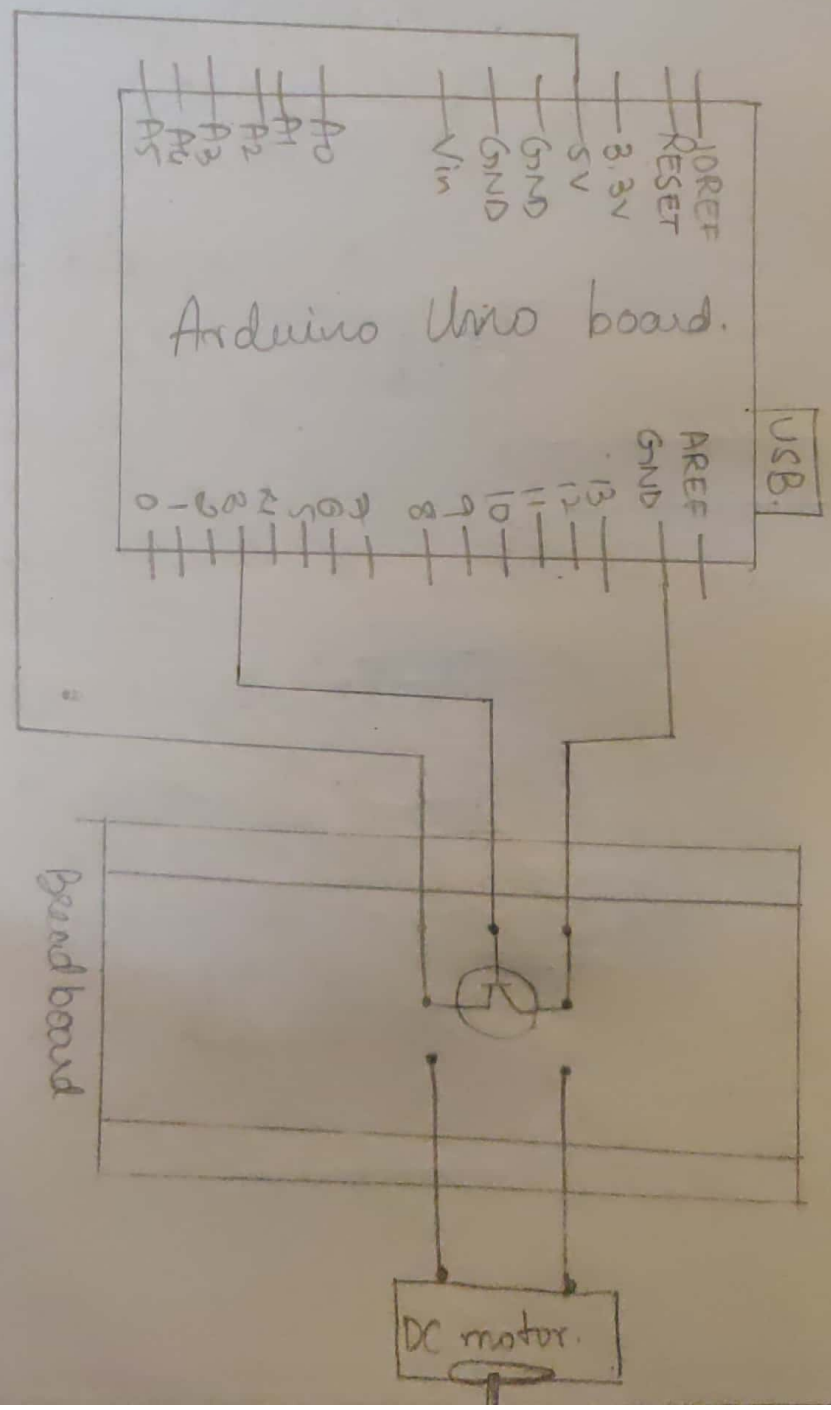
```
int m=3;
void setup()
{  pinMode (m, OUTPUT);
   Serial.begin (9600);
   while (!Serial);
   Serial.println ("Speed 0 to 255");
}
void loop()
{  if (Serial.available())
   {  int speed = Serial.parseInt();
      if(speed >=0 && speed <=255)
      {  analogWrite (m, speed); }
   }
}
```

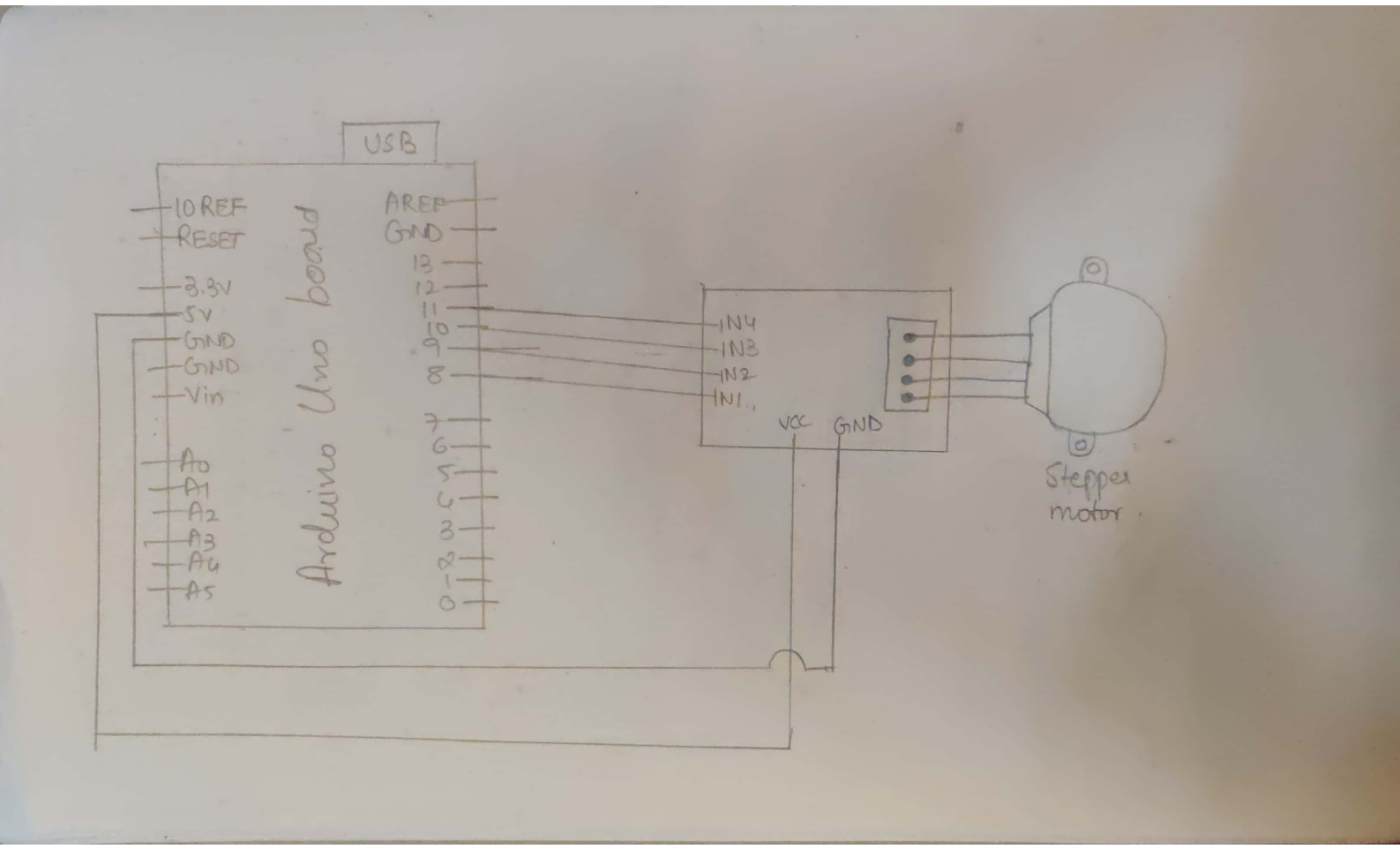
b) Stepper motor:

```
#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

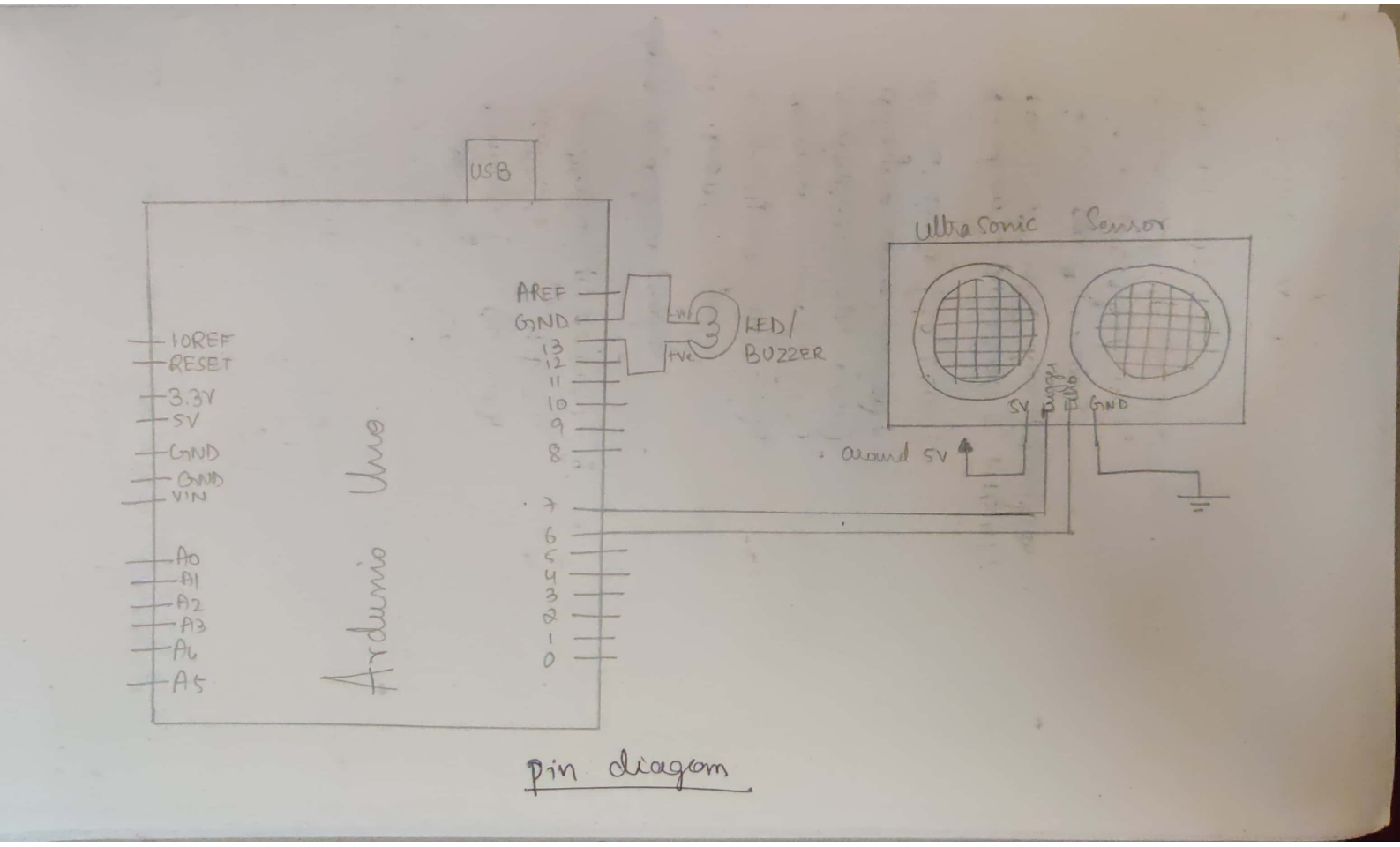




Program 8

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