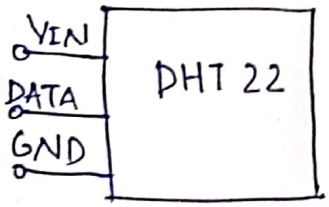
// code for transmitter end which transmits the data for a temp. and humd. sensor.

```
# include <VirtualWire.h> //for accessing VirtualWire func.
# include <DHT.h> // temp. & humd. Sensor func are under this header file
# define DHTPIN 4 // which pin collects the data from sensor
# define DHTTYPE DHT22 // specify which DHT sensor is used
const int led_pin = 13;
const int transmit_pin = 12; // pin used for transmission of data
 struct value
    {
      float tempreature; /* as the sensor gives two types of
      float humidity;          data i.e, tempreature and
    } data;                    humidity struct datatype is used */
  DHT dht (DHTPIN, DHTTYPE);
void setup()
{
  vw_set_tx_pin (transmit_pin); // specify the transmission pin
  vw_set_ptt_inverted (true); // configures ptt(push to talk)
  vw_setup (500); //initialises the library
  pinMode (led_pin, OUTPUT); // led to indicate when the data is transmitted
  Serial.begin(9600);
}
void loop()
  {
    [redacted]
    digitalWrite (led_pin, HIGH); //indicating the transmission process started
    readSensor();
    vw_send ((uint8_t *) & data, sizeof (data)); // sending the data with
    vw_wait_tx(); //wait until the whole          size of the data from
                  message is transmitted          sensor.
    digitalWrite (led_pin, LOW); //transmission process ended
    delay (2000); //delay of 2 seconds
  }
void readSensor()
{
  dht.begin(); // Initialisation of the sensor
  delay (1000); //delay of 1 second
  data.humidity = dht.readHumidity();
  data.tempreature = dht.readTempreature();
}
```

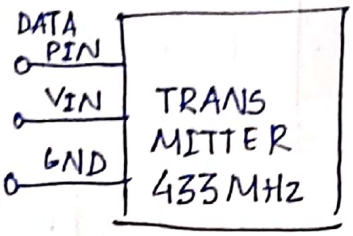
```
//code for the receiver
# include <VirtualWire.h>  // for accesing VirtualWire func.
const int receive_pin = 12;  //indicate receiver pin number.
const int led_pin = 13;
  struct value
    {
      float temprature = 0.0;
      float humidity = 0.0;
    } data;
void setup()
  {
    vw_set_rx_pin(receive_pin); //sets the receive pin
    vw_setup(500); //initialises the library
    vw_rx_start(); //start the receiver
    pinMode(led_pin, OUTPUT);
    Serial.begin(9600);
  }
void loop()
  {
    uint8_t buf[sizeof(data)];
    uint8_t buflen = sizeof(data);
      if (vw_have_message()) //Check whether there is a message
                             packet or not
       { digitalWrite(led_pin, HIGH); //data is being received
         vw_get_message(buf, &buflen);
         memcpy(&data, &buf, buflen);
         digitalWrite(led_pin, LOW);  //data reception ended.
         Serial.print("The temp. is:");
         Serial.println(data.temprature);
         Serial.print("\n The humidity is:");
         Serial.println(data.humidity);
       }
}
```

TRANSMITTER CKT. :→



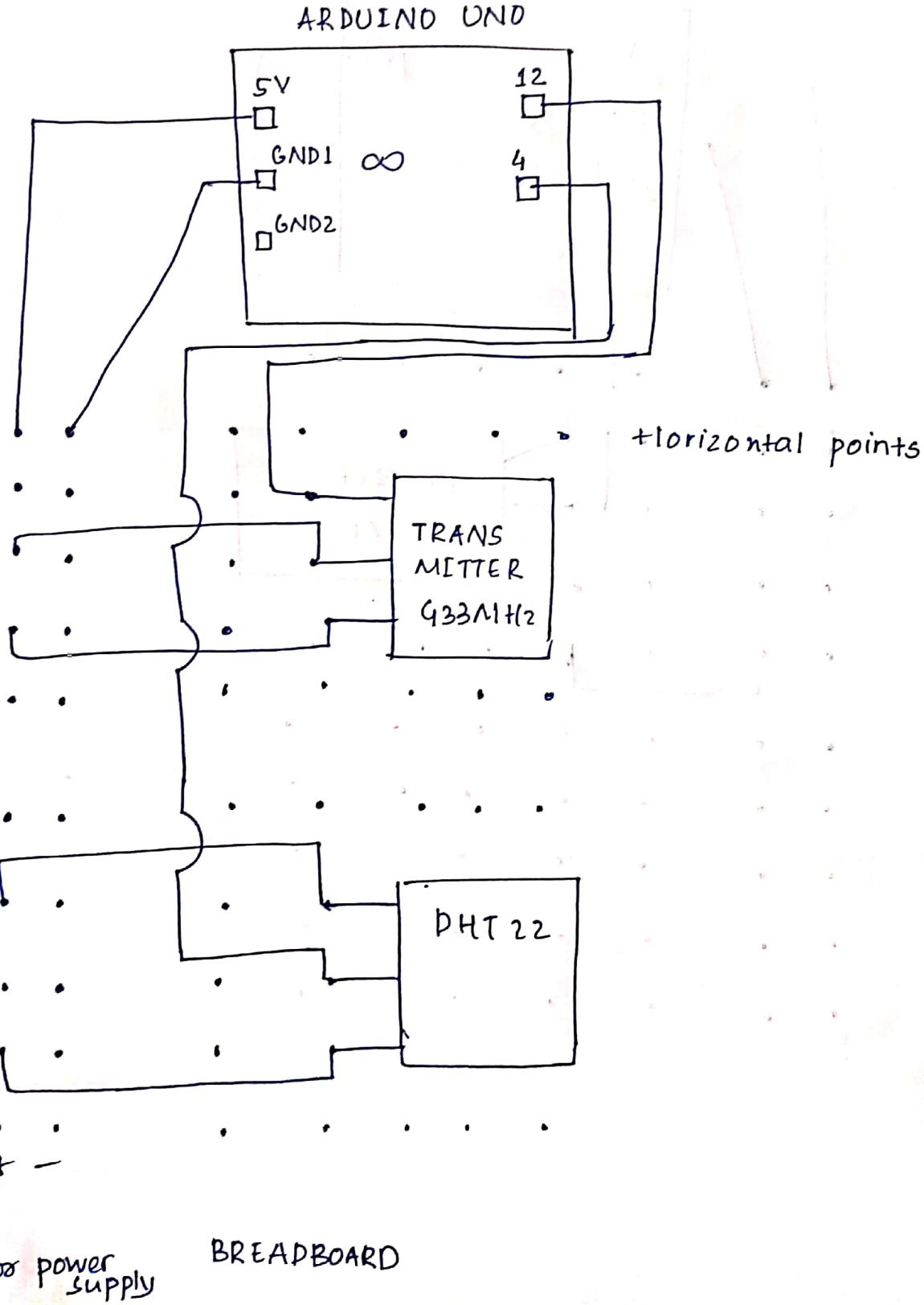
3 pins

(For tempreature and humidity measurement)



3 pins

(Tramitter module)



RECEIVER CKT. :→