

```
#include <Adafruit_Sensor.h>
#include <Adafruit_BMP280.h>
#include <DHT.h>

# define Hall sensor 2
#define DHTPIN 8
#define DHTTYPE DHT22 //MODELO DO SENSOR (DHT22 / AM2302)
DHT dht(DHTPIN, DHTTYPE); //Criar objeto DHT

Adafruit_BMP280 bmp; //OBJETO DO TIPO Adafruit_BMP280 (I2C)
float valor_hpa;
char serialData;
// Const def
float valor =0;
int Winddir =0;
int direcao[10];

// Constants definitions
const float pi = 3.14159265;
int period = 5000;
int radius = 147;
unsigned int counter = 0;
unsigned int RPM = 0;
float speedwind = 0;
float speedwind_nos = 0;

void setup(){
  pinMode(2, INPUT);
  digitalWrite(2, HIGH); //Ativa pull-up interno
  Serial.begin(9600);
  if(!bmp.begin(0x76)){ // ENDEREÇO I2C 0x76 do sensor
    Serial.println(F("Sensor BMP280 não foi identificado! Verifique as conexões."));
    while(1); //SEMPRE ENTRE NO LOOP
  }
  dht.begin();
}

float medirUmidade(){
  return dht.readHumidity();
}

float medirTemperatura(){
  return bmp.readTemperature();
}

float medirPressao(){
  valor_hpa = bmp.readPressure() / 100;
  return valor_hpa;
}

//direção vento
void realizaLeituraVento(){
  for(int i=0;i<10;i++)
  {
    valor = analogRead(A0) * (5.0 / 1023.0);

    if (valor <= 0.57) {
      Winddir = 315;
    }
    else if (valor <= 0.7) {
      Winddir = 270;
    }
    else if (valor <= 0.8) {
```

```

        Winddir = 225;
    }
    else if (valor <= 0.9) {
        Winddir = 180;
    }
    else if (valor <= 1.2) {
        Winddir = 135;
    }
    else if (valor <= 1.6) {
        Winddir = 90;
    }
    else if (valor <= 2.4) {
        Winddir = 45;
    }
    else {
        Winddir = 000;
    }

    direcao[i] = Winddir;
}
}

void bubbleSort (int dir[], int n) {
    int k, j, aux;

    for (k = 0; k < n - 1; k++) {
        for (j = 0; j < n - k - 1; j++) {
            if (dir[j] > dir[j + 1]) {
                aux = dir[j];
                dir[j] = dir[j + 1];
                dir[j + 1] = aux;
            }
        }
    }
}

void loop(){
    if(Serial.available() > 0){
        serialData = Serial.read();

        if(serialData == 't'){
            Serial.println(medirTemperatura());
        }

        if(serialData == 'p'){
            Serial.println(medirPressao());
        }

        if(serialData == 'u'){
            Serial.println(medirUmidade());
        }

        if(serialData == 'd'){
            realizaLeituraVento();
            bubbleSort(direcao,10);
            Serial.println(direcao[5]);
        }

        if(serialData == 'a'){
            windvelocity();
            RPMcalc();
            SpeedWind();
            speedwind_nos = speedwind/1.8;
            Serial.println(speedwind_nos);
        }
    }
}

```

```
    }

}

// Measure wind speed
void windvelocity(){
    speedwind = 0;

    counter = 0;
    attachInterrupt(0, addcount, RISING);
    unsigned long millis();
    long startTime = millis();
    while(millis() < startTime + period) {
    }
}

void RPMcalc(){
    RPM=((counter)*60)/(period/1000);
}

void SpeedWind(){
    speedwind = (((4 * pi * radius * RPM)/60) / 1000)*3.6;
}

void addcount(){
    counter++;
}
```