Corrected DIY Ventilator + Pulse Oximeter Code

```
// ====== LIBRARIES ======
#include <MsTimer2.h>
#include <LiquidCrystal.h>
#include <Wire.h>
#include <mwc_stepper.h>
#include <SoftwareSerial.h>
#include "MAX30100_PulseOximeter.h"

// ====== PIN DEFINITIONS ======
#define EN_PIN 0
#define DIR_PIN 7
#define STEP_PIN 6
#define buzzer 2

#define adjust_speed_knob A0
#define adjust_angle_knob A3
#define set_speed_key 3
#define set_angle_key 4

#define PULSE 3000
#define CLOCKWISE 0
#define ANTICLOCKWISE 1

// ====== OBJECTS ======
MWCSTEPPER nema23(EN_PIN, DIR_PIN, STEP_PIN);
LiquidCrystal lcd(8, 9, 10, 11, 12, 13);
SoftwareSerial atmega_master(A2, A1);
SoftwareSerial atmega_slave(3, 2);
PulseOximeter pox;

// ====== GLOBAL VARIABLES ======
volatile int Addr = 0x78;
volatile float pressure = 0.00;
int RPM = 200;
int step_angle = 3000;
uint32_t tsLastReport = 0;
int heart_rate;
int spo2;

// ====== SETUP ======
void setup()
{
  Serial.begin(9600);
  Wire.begin();
  lcd.begin(20,4);
  nema23.init();

  pinMode(buzzer, OUTPUT);
  pinMode(set_speed_key, INPUT_PULLUP);
  pinMode(set_angle_key, INPUT_PULLUP);

  lcd.print("DIY Ventilator");
  delay(1000);
  lcd.clear();

  // Pulse Oximeter Init
  if (!pox.begin())
    Serial.println("Pulse Oximeter FAILED");
  else
    Serial.println("Pulse Oximeter SUCCESS");

  pox.setIRLedCurrent(MAX30100_LED_CURR_7_6MA);
}

// ====== LOOP ======
void loop()
{
  stepper_motor_control();
  measure_air_pressure();

  // Pulse Oximeter Update
  pox.update();
```

```
  if (millis() - tsLastReport > 1000)
  {
    heart_rate = pox.getHeartRate();
    spo2 = pox.getSpO2();

    Serial.print("HR: ");
    Serial.print(heart_rate);
    Serial.print("  SpO2: ");
    Serial.println(spo2);

    tsLastReport = millis();
  }
}

// ====== STEPPER MOTOR CONTROL ======
void stepper_motor_control()
{
  nema23.set(CLOCKWISE, RPM, PULSE);
  for (int i=0; i<step_angle; i++)
    nema23.run();

  delay(200);

  nema23.set(ANTICLOCKWISE, RPM, PULSE);
  for (int i=0; i<step_angle; i++)
    nema23.run();

  delay(200);
}

// ====== PRESSURE SENSOR FUNCTION ======
void measure_air_pressure()
{
  Wire.requestFrom(Addr, 4);
  if (Wire.available() == 4)
  {
    int data0 = Wire.read();
    int data1 = Wire.read();
    int raw = (data0 * 256) + data1;

    float psi = ((raw - 3277.0) / (26214.0 / 1.5));
    pressure = psi * 6894.76;
    pressure = pressure * 0.0101972 * 2.8;

    Serial.print("Pressure (cmH2O): ");
    Serial.println(pressure);
  }
}
```