Application Note AN144:
K-30 CO2 Sensor Troubleshooting with Arduino or Raspberry Pi

Customers using SenseAir's K-30 CO2 sensor with an Arduino micro controller or the Raspberry Pi occasionally see the sensor "lock up" during use. It appears that the sensor sends negative data and quits working. After unplugging the sensor from power momentarily, it works again. This seems to be a random problem.

Fortunately, we have been able to recreate the problem, and have 5 possible solutions.

1. Low Power

The most common issue is power. 5VDC is the absolutely required minimum voltage. If you are powering from a USB port, the nominal 5.25V will drop with other USB devices on the circuit, causing an error. If the sensor detects low voltage when it enables the source lamp (peak current about 150mA), it terminates the measurement cycle and reports 32767/7FFF or negative data as the measurement value. We have been able to see this using an oscilloscope.

If you do not have consistent 5.25VDC or greater power from your PC or laptop's USB connector, the solution is to use an external powered USB hub or 6VDC power supply.

If you verify you have enough power via USB, you can try adding a 47uF capacitor at the power input of the sensor. This may increase available power when the peak current is requested from the IR lamp.

2. Exception Handling

Another reason we have seen sensors lock up is if the software does not test for or know how to handle exceptions when the request for the measured value return an incorrect CRC. These errors are generally the result of software timing or intermittent spikes caused by faulty wiring.

Our source code to connect the K-30 sensor to the Arduino or Raspberry Pi does not handle every exception. It is up to you to add exception handling to your code.

3. Polling Sensor every 2 Seconds or Less

The third reason we have seen lockups is polling too often. Note that while you can poll the sensor as often as you want, measurements on the K30 are made every 2 seconds. Polling the device more than this will not make more accurate readings, and can result in lockups if the software does not have proper error handling. **4. Poor TX/RX Signal**

Another reason for lockups is that the host device may have conditioning (pull ups or lack of pull ups) that make the RX or TX signal swing marginal. You can use an oscilloscope to verify that the signals have good highs and lows. **5. Noise from other Devices**

Finally, in this case of the transient potential, look for unclamped radiated noise from inductive switching, i.e. motors or solenoids. You may have to install clamps on the sensor power and or communications lines if you cannot control the source.