Application Brief

Leveraging Relative Humidity Sensor Enhanced Features for Ultra-Low-Power Systems

TEXAS INSTRUMENTS

Christy She

Supply current is an important selection criteria for sensors in many systems, especially battery-powered systems or enclosed systems trying to avoid measurement errors from self heating. It is often not as simple as taking a single number from the data sheet, especially for slower duty-cycled sensing applications where temperature and humidity are collected. In that case, both the sleep and active current must be used along with the measurement frequency and measurement duration to calculate the total or average current drawn. Features of the sensor, such as an alert pin which goes high when the sensed value is outside a pre-programmed threshold, can reduce system-level current for some applications by allowing the higher current MCU or DSP processor to stay asleep longer and only be woken by the alert pin of the sensor when the sensed value is outside a preprogrammed threshold.

This paper focuses on relative humidity sensors and shows how to calculate the average sensor current, options to reduce current, and looks at system-level current considerations to achieve the lowest current for your intended application.

Calculating Average Sensor Current

For any duty-cycled sensor, one that goes to sleep between measurements, use Equation 1 to calculate its average current consumption.

$$I_{sensor\_avg} = \left(f_m \times I_{meas} \times t_{meas}\right) + \left(I_{sleep} \times \left(1 - f_m \times t_{meas}\right)\right) \quad (1)$$

where

- f_m is the measurement frequency in Hz
- I_{meas} is the supply current while the sensor is actively measuring. This is often referred to as I_{active} in RH sensor data sheets.
- I_{sleep} is the supply current while the sensor is asleep (between measurements)
- t_{meas} is the active measurement duration in seconds

Is Sleep Current or Active Current More Important for a Low-Power Sensor?

Many sensing modalities, such as humidity and temperature, measure at a low frequency which allows the sensor to go into a low-power (sleep) state between measurements. Some sensors, such as the HDC3020, automatically enter this sleep mode between measurements but other sensors require a specific command to enter sleep mode. The lower the measurement frequency, the larger the benefit of sleep-mode current reduction because more time is spent there and less in a higher current measurement state.

Figure 1 illustrates the calculated average measurement current (using Equation 1) across measurement frequency for the HDC3020 relative humidity (RH) and temperature sensor. While the slope and offset is specific to the HDC3020 sensor, the trend of average current increasing as sensing frequency increases applies to all sensors. This is because the more measurements per second, the more time the sensor spends in the higher current active measurement state verses sleep. Thus, depending on the measurement frequency of the application, either sleep or active current could become a bigger contributing factor to the overall average current.

To enable a lower current consumption profile, humidity and temperature sensors often offer the selection of a lower power mode when the lowest noise is not required for the application. Various names for this configuration are used across different sensors. Some popular names include: resolution accuracy, low-power mode, and repeatability. When selecting a lower power mode often the measurement duration decreases and the active current may also decrease. Figure 1 illustrates the current savings. All four power modes have very similar average currents at the slowest measurement frequency. This is because at less than 1-Hz measurement frequency the average current is dominated by sleep current which does not change across the modes. However, as the measurement frequency increases, the current delta between modes increases as the active current becomes more dominant.

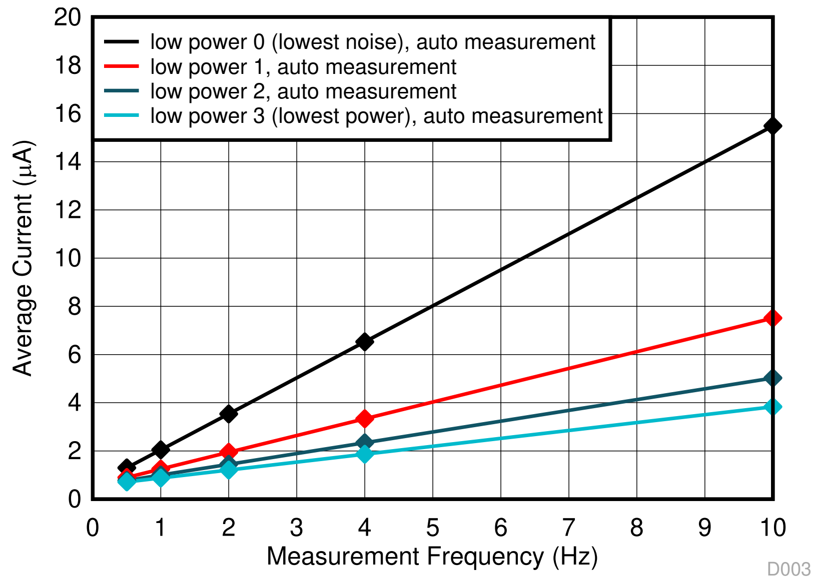

Figure 1. HDC3020 Average Humidity Sensor Current vs Measurement Frequency for Different Power Modes

Figure 2 plots the average current vs measurement frequency for the HDC2080 and HDC3020 humidity and temperature sensors. HDC2080 has almost 10 × lower sleep current but HDC3x has 6 × lower active current. The graph shows that at 1 Hz and higher measurement frequency, the lower active current of the HDC3x dominates with having the lowest average current. At very low measurement frequencies, 0.5 Hz and slower, the lower sleep current of the HDC2080 results in a lower current. This shows the lowest power sensor depends on the application measurement frequency.

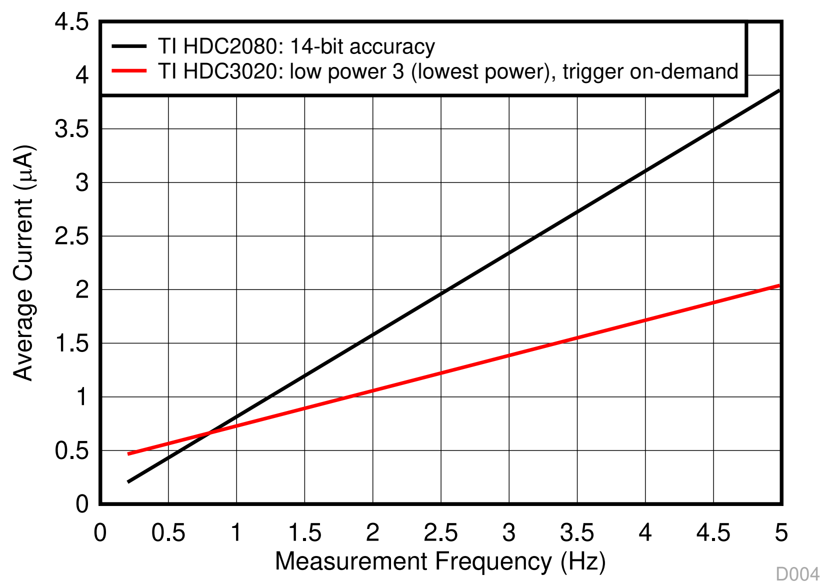

Figure 2. Average Humidity Sensor Current vs Measurement Frequency for HDC2080 and HDC3020

Comparing humidity sensor current between devices may not be as easy as pulling a single number from the data sheets especially when comparing different vendors. Often the average current for 1 measurement per second is provided in the data sheet so if this is the measurement frequency of the application and the test conditions apply to your application the average current does not have to be calculated, but pay close attention to the test conditions. For Figure 2, the highest resolution accuracy for HDC2080 was selected to compare to the HDC3020 lowest power mode as those modes have similar repeatability and measurement noise.

Figure 3 shows the HDC3020 vs Competitor A. Both devices have the same RH accuracy specification: ±1.5% typical and ±2% maximum RH accuracy. Competitor A data sheet only specifies average current for low repeatability so calculations were made using Equation 1 and data sheet values to compare. Competitor A high repeatability mode has similar repeatability as HDC3020 low-power mode 3 so those settings were selected to compare the two devices. In continuous measurement mode (called auto measurement and periodic mode on the two devices), HDC3020 is approximately 60 times lower current than Competitor A at 1 measurement per second and in single shot or trigger on-demand mode HDC3020 is 10 × lower average current than Competitor A at 1 measurement per second. Even if the HDC3020 is configured to low power mode 0, which has higher repeatability (less noise in the measurement) than Competitor A, HDC3020 is still much lower current than Competitor A: 4 × lower in single shot and 25 × lower in continuous measurement mode.

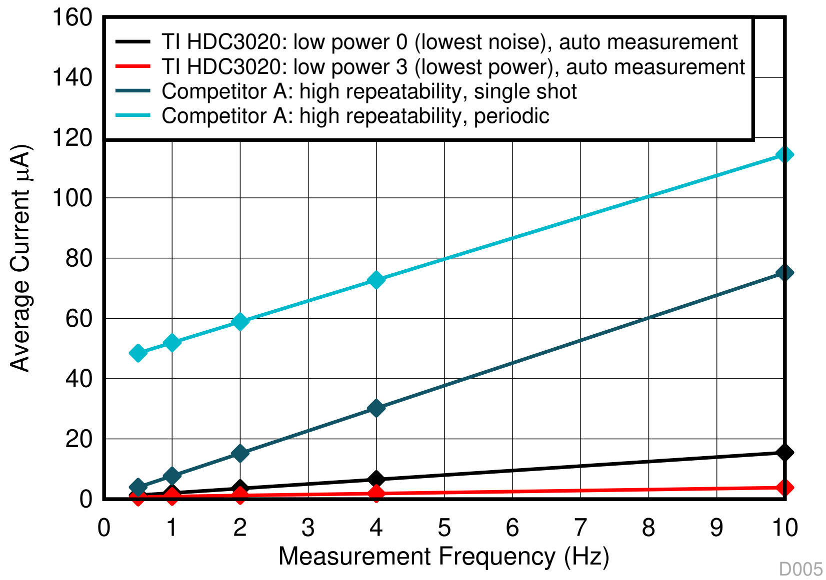

Figure 3. Competitive Comparison Average Current vs Measurement Frequency

Notice from Figure 3 that even though competitor A has 2 × lower sleep current than HDC3020, the HDC3020 is always lower current because its 1/6 times lower active current dominates the average current. In contrast, the periodic mode of Competitor A has almost 10 × higher sleep current so that dominates at the lower measurement frequencies and because Competitor A has higher active current too, it remains the higher current across measurement frequencies.

System-Level Current Implications of the Features of a Sensor

Battery-powered applications care about battery lifetime because batteries dying lead to the inconvenience of battery changes for consumers or costly maintenance for companies to replace batteries. To maximize battery lifetime, the system-level current needs to be minimized to budget the lifetime of the battery. Current from the humidity sensor, I2C communication, a processor in the system to read the humidity sensor are among the system-level considerations. At the system-level features such as auto measurement mode, data ready interrupt, and alert pin make an impact on total system current as they allow the processor to remain in a low-power state longer.

To consider system-level current for a humidity sensor, Equation 1 needs to be expanded to include I2C communication current for requesting a measurement as well as to read the measurement values. These numbers are generally not provided in the data sheet but can be obtained from bench testing with a scope trace to measure the supply current to the humidity sensor and the I2C pullup resistors currents. A pullup resistor on the I2C SCL pin is only required if the host processor pin is open-drain. The I2C SDA pin requires a pullup resistor and needs to be sized to draw more current for longer communication distances so is application dependent. The current from the processor that communicates with the humidity sensor also needs to be considered both while it is active for I2C communication as well as while it is waiting for sensor data. The average system current is given in Equation 2 with the detailed expansion of the equation in Equation 3.

$$I_{system\_avg} = I_{sensor\_avg} + I_{I2C\_avg} + I_{MCU\_sleep\_avg} \tag{2}$$

where

- I_{sensor_avg} is defined in Equation 1
- I_{I2Cavg} is the current associated with I2C averaged out over time
- $I_{MCU\_sleep\_avg}$ is the current for MCU in lowest power mode waiting for the next sensor measurement or read request averaged out over time

$$I_{system\_avg} = I_{sensor\_avg} + \left(f_m \times \left(I_{sensorI2C} + I_{MCU\_active} + I_{MCU\_I2C} + \left(I_{SDA} \times SF_{SDA}\right) + \left(I_{SCL} \times SF_{SCL}\right)\right) \times \left(t_{I2Cmeas} + t_{I2Cread}\right)\right) + \left(I_{MCU\_sleep} \times \left(1 - \left(f_m \times \left(t_{I2Cmeas} + t_{I2Cmeas}\right)\right)\right)\right) \tag{3}$$

where

- I_{sensor_avg} is defined in Equation 1
- f_m is the measurement frequency in Hz
- $I_{sensorI2C}$ is the humidity sensor current during I2C communication
- I_{MCU_active} is the current for MCU in lowest power mode that allows for I2C communications, or whatever power mode the application requires
- I_{MCU_I2C} is the current for the MCU I2C peripheral
- I_{SDA} is the current through the I2C SDA pin pullup resistor. This is simply V_{DD}/R_{SDA_pullup}.
- SF_{SDA} is the scaling factor which represents the % of I2C data that is zero because the SDA pullup resistor only sources current during data low.
- I_{SCL} is the current through the SCL pullup resistor. This is simply V_{DD}/R_{SCL_pullup}. This is only required if the host processor pin is open-drain, else if the pullup resistor is not present, use the value 0 to remove this term.
- SF_{SCL} is the scaling factor which represents the % of time SCL is low. If SCL clock duty cycle is 50%, this is 0.5
- $t_{I2Cmeas}$ is the duration of I2C communication to request measurement in seconds
- $t_{I2Cread}$ is the duration of I2C communication to request and read humidity and temperature data in seconds
- I_{MCU_sleep} is the current for MCU in lowest power mode waiting for the next sensor measurement or read request

I_{system_avg} does not include any application-level current outside of interacting with the humidity sensor itself. Other current-consuming activities might include displaying readings on an LCD or any RF communication which can quickly dominate the average current unless data transmission is sparse either by infrequent data log transmission or by local processing and decision making to limit data transmission to conditions requiring broader action.

Several common humidity sensor features can lower system current by allowing the MCU to remain in sleep mode longer or enter into lower current (deeper) sleep mode and reducing required I2C communication. The features offered on the HDC3020 family are: auto measurement mode, data ready interrupt, and ALERT.

The auto measurement mode of a humidity sensor enables it to automatically measure humidity and temperature at a user-selected measurement frequency. Current is saved as no I2C communication is required to request the measurement and the MCU can stay in low-power mode where only a low power timer is running to tell it to periodically wake up to read the sensor data. If the data ready interrupt is used, even more current is saved because the MCU can be in deepest sleep (wake upon interrupt) until receiving the data ready interrupt. Auto measurement mode often has a higher humidity sensor sleep current than trigger on demand, a single measurement, because it needs to run a timer to automatically trigger the next measurement. Thus, calculate with Equation 3 to determine if and how much system-level current is saved in this mode as the system-level current saved is application-specific.

If the MCU only needs to monitor humidity and take action when a certain RH percentage is reached, the alert function of the humidity sensor can take over the monitoring task and wake up the MCU once the condition is met. The HDC3020 ALERT feature allows the user to set minimum and maximum thresholds for both RH and temperature to be monitored. The humidity sensor monitors the temperature and RH measurements and if they fall outside the thresholds, the ALERT pin is set high. This smart processing within the humidity sensor allows the MCU to remain in the deepest sleep mode until receiving an alert to wake it up versus reading the sensor after every measurement and checking if the measurement is out of range. This alert feature reduces the total average system current with the savings being the most for the highest-frequency auto measurement mode settings.

Table 1 summarizes what system-level currents play a role for different combinations of system-level current savings features. This quickly shows which currents play a role in system current of which use case. N/A stands for *not applicable* and means that current is not used in that use case. I_{SCL} is only a factor when a pullup resistor is used on the I2C SCL pin.

Table 1. System-Level Average Humidity Sensor Current Consideration

Use Case	$I_{sensorI2C}$, I_{SDA}, I_{SCL}	$t_{I2Cmeas}$	$t_{I2Cread}$	I_{MCU_sleep} wake on pin interrupt	I_{MCU_sleep} with RTC timer	I_{MCU_active}	I_{MCU_I2C}
Single Read	Y	Y	Y	N/A	Y	Y	Y
Auto Measurement	Y	N/A	Y	N/A	Y	Y	Y
Auto Measurement with Data Ready	Y	N/A	Y	Y	N/A	Y	Y
ALERT	N/A	N/A	N/A	Y	N/A	N/A	N/A

Conclusion

There is a lot to consider when looking for a low-power humidity sensor. Both standby and active current can matter as well as sensor modes to reduce current. Also, it is critical to consider overall system-level current when calculating the power budget of the system, and features like auto measurement or alert on the humidity sensor can make a big impact on lowering system-level current for some applications. The HDC3020 humidity sensor has approximately 10 × lower current than the competition in its accuracy class at 1 measurement per second in single-shot mode and approximately 60 × lower current in auto-measurement mode. HDC3020 also offers a data ready interrupt and alert feature to reduce system-level current for many applications. If your applications require high accuracy humidity sensing and low current, first consider Texas Instruments' HDC3020 and HDC3020-Q1, the highest accuracy and lowest power humidity and temperature sensor in the industry.