Air Quality Egg Sensor Shield Specification

Table of Contents

Overview

The Air Quality Egg Sensor Shield (AQESS) achieves the following primary objectives:

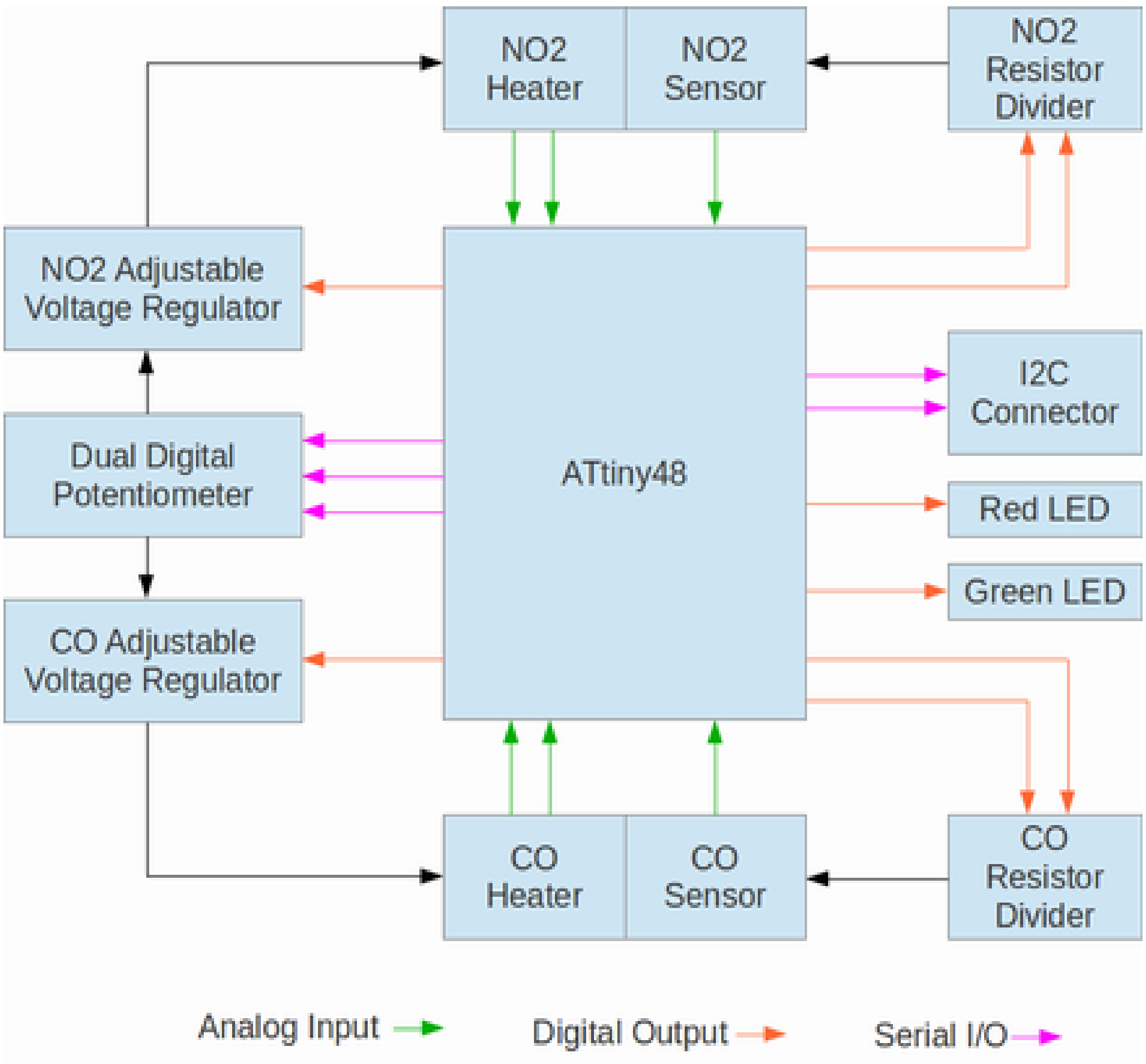
- Hosts an NO_2 Gas Sensor (e2v model MiCS-2710) and a CO Gas Sensor (e2v model MiCS-5525)
- Provides hardware for an add-on sensor expansion capability
- Provides a well-defined interface for the Nanode (or other host board) to access the hosted gas sensors and any add-on sensors

The remainder of this document describes the details of how these primary objectives are implemented.

Gas Sensor Hosting

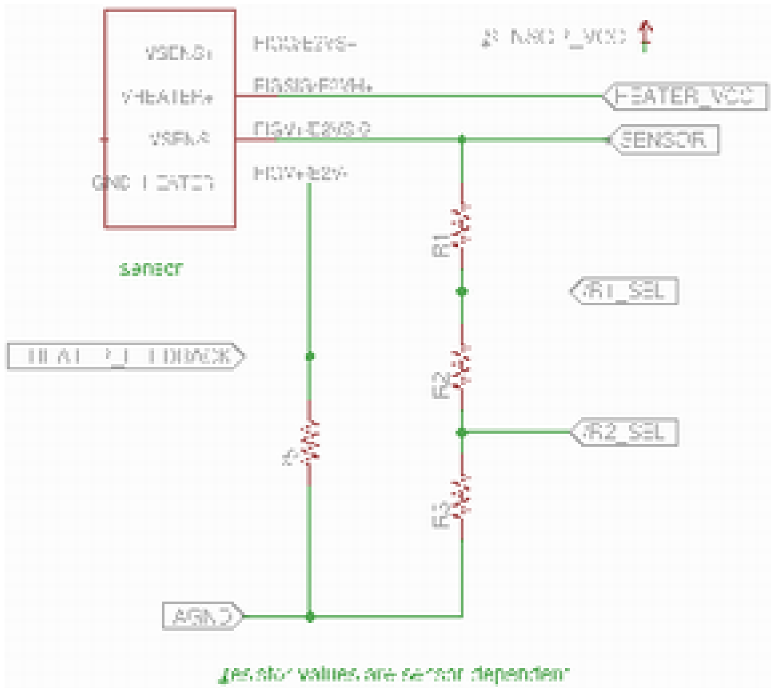
Overview

The following block diagram summarizes the monitoring and control of the gas sensors at a high level.



Two sockets are provided; one for the NO_2 Gas Sensor (e2v model MiCS-2710) and one for the CO Gas Sensor (e2v model MiCS-5525). The sockets themselves are visually indistinguishable, but the printed circuit board is labelled to make clear which sensor belongs in which socket.

The circuits presented throughout this section may not be exact excerpts from the project schematics, and so the designators may be different, but the circuits are representative and analogous to those implemented. The sub-circuit for CMOS Gas Sensors is as follows:



Each gas sensor has its own operating voltage (SENSOR_VCC) and heater power requirement. These are as follows per the sensor datasheets.

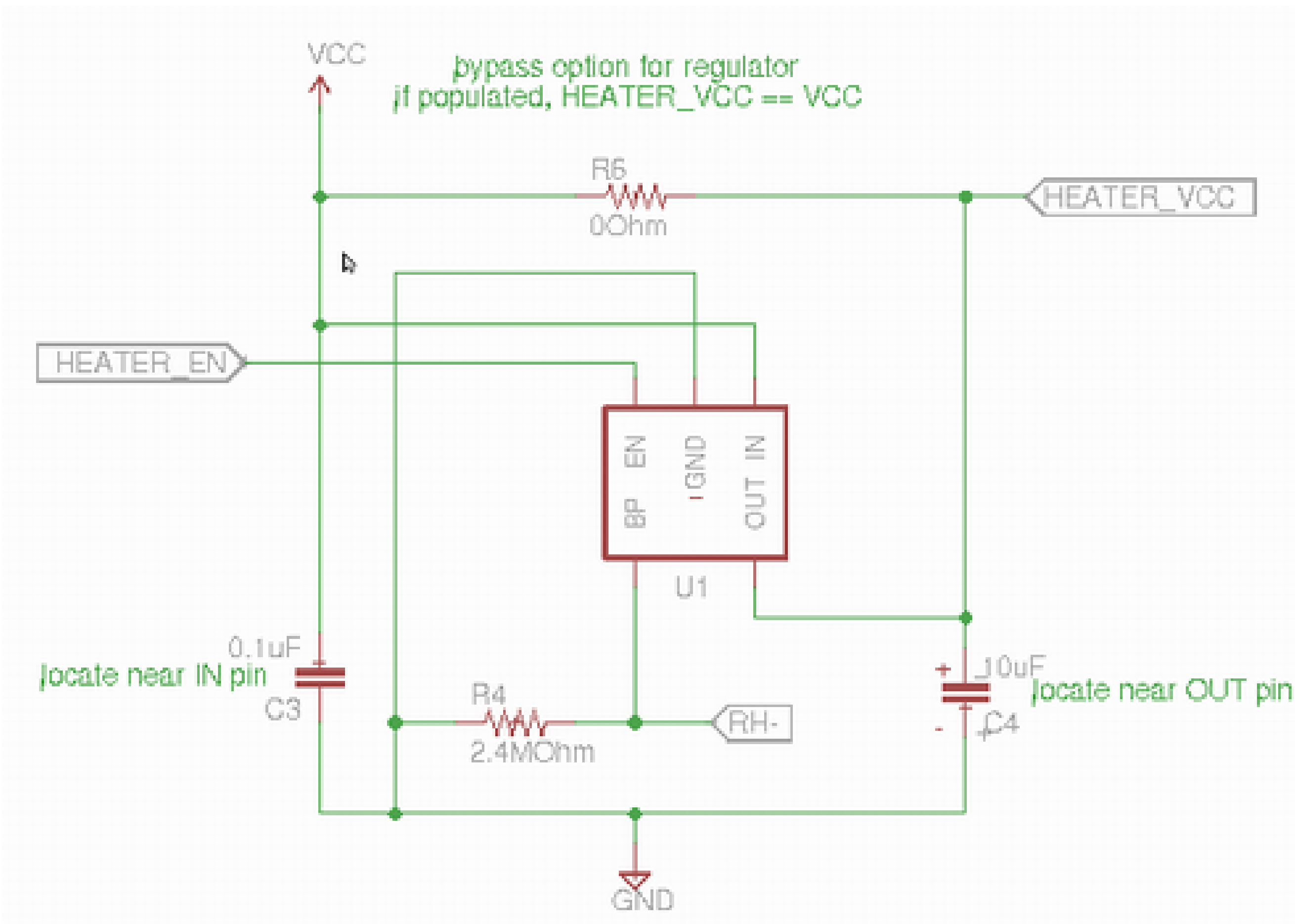
Table 1. Sensor Operating Parameters

Sensor	Operating Voltage	Heater Power
MiCS-2710 (NO_2)	2.5 V	43 mW
MiCS-5525 (CO)	5.0 V	76 mW

The regulated 5.0 V power from the Nanode will provide the operating voltage for the CO Gas Sensor, and a dedicated 2.5 V linear regulator will provide the operating voltage for the NO_2 Gas Sensor.

Heater Control

The power dissipated in each sensor heater will be controlled through a feedback control loop. The Heater supply voltage (HEATER_VCC) is generated by an adjustable voltage regulator (Microchip TC1071VCT713), with the following sub-circuit:



A digitally controlled potentiometer is connected between the RH- and HEATER_VCC terminals in the circuit above, whose variable resistance will be denoted R_{POT}. The voltage output by the adjustable regulator is related to R4 and R_{POT} by the following equation:

$$HEATER\_VCC = 1.2 \cdot \left(\frac{R_{POT}}{R4} + 1 \right)$$

The range of voltages that can be generated by this circuit is 1.2 V to VCC (i.e. 5.0 V). Importantly, HEATER_VCC varies linearly with R_{POT} in this formula. The slope of the line relating HEATER_VCC to R_{POT} is (1.2 / R4) V/Ohm. In order to achieve 5V output, R_{POT} must be capable of being set to at least 3.17 times the value of R4.

The power dissipated in the heater is calculated as the voltage across the heater multiplied by the current through the heater. The current through the heater is precisely the voltage measured at the node labelled HEATER_FEEDBACK divided by the resistance of R5. The voltage across the heater is precisely the difference between HEATER_VCC and HEATER_FEEDBACK. Combining this analysis yields the following equation:

$$HEATER\_POWER = \left(HEATER\_VCC - HEATER\_FEEDBACK\right) \cdot \frac{HEATER\_}{R}$$

$$HEATER\_POWER = \left[1.2 \cdot \left(\frac{R_{POT}}{R4} + 1\right) - HEATER\_FEEDBACK\right] \cdot \frac{HEATER\_FEE}{R5}$$

By measuring, HEATER_VCC and HEATER_FEEDBACK, and knowing the value of R5 precisely, the power being dissipated in the heater at any moment can be calculated accordingly. Knowing the target power for either sensor, R_{POT} can thereby be iteratively adjusted to keep the measured power dissipation in range over time, independent of drift in resistive characteristics of the heating element over the lifetime of the sensors. In summary, to increase power dissipation R_{POT} must be increased, and to decrease power dissipation R_{POT} must be decreased. This monitoring and feedback control process is managed by a dedicated microcontroller (ATtiny48) which can manage the two sensors hosted on the shield in parallel, while also providing a digital interface to the Nanode reading the current sensor values.

Sensor Measurement

The measurement of the sensor response is implemented by using the microcontroller's 10-bit analog to digital converter (ADC). The sensor resistance (which varies with exposure to pollutants) forms a voltage divider between either R1, R1 + R2, or R1 + R2 + R3 depending on the state of /R1_SEL and /R2_SEL. This allows the microcontroller to adjust the range of values it can measure with improved resolution, and to cope with the large range of resistances that the full scale response of the sensors cover. It is necessary to do use this technique because of the inherently nonlinear response of CMOS Gas Sensors. The following table describes how the lower leg of the voltage divider is configured with respect to /R1_SEL and /R2_SEL.

Table 2. Sensor Voltage Divider Settings

/R1_SEL	/R2_SEL	R_LOW_SIDE
High-Impedance	High-Impedance	R1 + R2 + R3
High-Impedance	GND	R1 + R2
GND	High-Impedance	R1

By measuring the voltage at the midpoint of the voltage divider (SENSOR) and knowing the sensor operating voltage and the resistance of the low side of the divider (R_LOW_SIDE), the resistance of the sensor (R_{SENSOR}) can be computed by means of the standard voltage divider equation:

$$SENSOR = SENSOR\_VCC \cdot \frac{R\_LOW\_SIDE}{R\_LOW\_SIDE + R_{SENSOR}}$$

$$R_{SENSOR} = R\_LOW\_SIDE \cdot \left(\frac{SENSOR\_VCC}{SENSOR} - 1 \right)$$

This formula can subsequently be converted in terms of Analog-to-Digital conversion counts by way of the following relation (for a 10-bit ADC):

$$ADC = \frac{SENSOR}{ADC\_VCC} \cdot 1024$$

$$\frac{ADC\_VCC}{SENSOR} = \frac{1024}{ADC}$$

$$\frac{ADC\_VCC}{SENSOR} \cdot \frac{SENSOR\_VCC}{ADC\_VCC} = \frac{1024}{ADC} \cdot \frac{SENSOR\_VCC}{ADC\_VCC}$$

$$\frac{SENSOR\_VCC}{SENSOR} = \frac{1024}{ADC} \cdot \frac{SENSOR\_VCC}{ADC\_VCC}$$

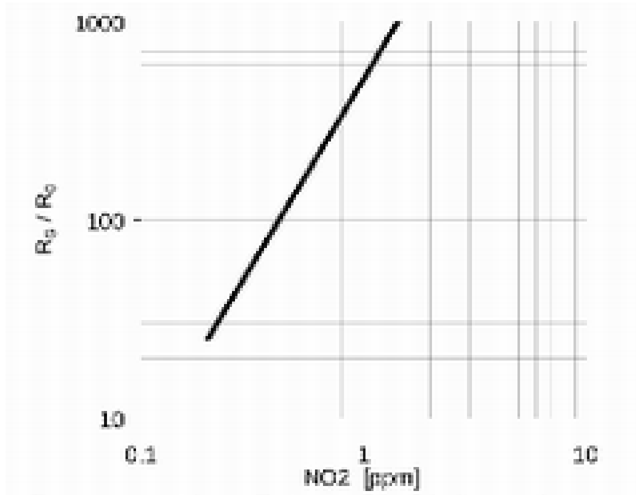
By substitution into the previous equation for R_{SENSOR} we arrive at the following result:

$$R_{SENSOR} = R\_LOW\_SIDE \cdot \left(\frac{1024}{ADC} \cdot \frac{SENSOR\_VCC}{ADC\_VCC} - 1 \right)$$

$$R_{SENSOR} = \frac{R\_LOW\_SIDE \cdot SENSOR\_VCC}{ADC\_VCC} \cdot \left(\frac{1024}{ADC} - \frac{ADC\_VCC}{SENSOR\_VCC} \right)$$

$$\frac{R_{SENSOR}}{R0} = \frac{R\_LOW\_SIDE \cdot SENSOR\_VCC}{R0 \cdot ADC\_VCC} \cdot \left(\frac{1024 \cdot SENSOR\_VCC - ADC\_VCC \cdot ADC}{ADC \cdot SENSOR\_VCC} \right)$$

This formula determines the sensor resistance in terms of quantities we can control (R_LOW_SIDE), quantities we can measure directly (ADC), and constants we know by design (R0, ADC_VCC, SENSOR_VCC). Having calculated the resistance of the sensor, we can use that value to extrapolate to a gas concentration from a table stored in the microcontroller (or in peripheral memory) that encodes the sensor response curve from its datasheet such as the following (from the MiCS-2710 datasheet):



The sensor resistance R_S is normalised to the resistance under air (R_0).

From this curve, along with the indication in the datasheet that R_0 is nominally 2.2 kΩ, we would choose R1 = 2.2 kΩ, R2 = 22 kΩ, and R3 = 220 kΩ.

Egg Bus Interface Definition

Overview

The host controller’s (e.g. Nanode’s) abstraction of a Sensor Module is a well defined Memory Map. This Memory map can be accessed (read and written) over a standard I2C bus (aka TWI bus), and the command / response protocol along with the Memory Map defined herein defines the Egg Bus Interface Definition. Any module that supports the I2C commands and expected responses as defined in this section should be pluggable into the Air Quality Egg system.

At a high level, the Memory Map is composed of a small header section containing metadata which all Sensor Modules have, followed by a sequence of Sensor Data Blocks depending on how many sensors are provided by that device. By convention, all multi-byte values in the Memory Map are transmitted on the bus most-significant-byte first (i.e. big-endian / network byte order). When requesting an address from the table below, the expectation is that the Nanode will read the indicated number of bytes in total.

Command Response Protocol

In order to read a value from the Memory Map, the Nanode must write the target Sensor Module’s I2C address to bus with the Write bit set to 0 (SLA+W), then write the “READ” command to the bus (0x11), then write the address to be read to the bus (high byte then low byte), and finally an I2C stop condition. The Nanode then writes the Sensor Module’s I2C address to bus with the Write bit set to 1 (SLA+R), and clocks in the expected number of bytes as expected based on the address requested, and finally issues an I2C stop condition. To minimize decoding complexity most addresses identify four byte integer values, with the exception of locations of ASCII strings which are conventionally 16-byte null-terminated strings.

Address	Size (Bytes)	Offset	Field Name	Description
0	1	0	Sensor Count	The number of sensor blocks provided by this device
1	6	1	Module ID	6-byte Unique Sensor Module ID
7	4	7	Version	Version Number of Firmware
9	21	11	SPARE	Space for growth
				*The following **Sensor Block** address range is repeated once for each supported sensor*
32	16	0	Sensor Type	Null-terminated ASCII String (at most 15 characters). E.g. "NO2"
48	16	16	Sensor Units	Null-terminated ASCII String (at most 15 characters). E.g. "ppb"
64	4	32	Sensor R0	Sensor resistance, measured in units of ohms, in air under controlled ambient conditions.
68	4	36	Measured Independent Variable	The un-mapped value to which the Bounding Box can be applied to arrive at a value in units returned by Sensor Units, represented as a 32-bit floating point value.
72	4	40	Table X Scaler	32-bit floating point value
76	4	44	Raw Sensor Value [Optional]	Atomically provides two 2-byte values. The first value is the 10-bit ADC reading. The second value is the resistance of the low side of the voltage divider in hundreds of ohms (e.g. a value of 22 would be reported for 2200 = 2.2k ohms). The sensors are presumed to use the 5V reference voltage for the ADC conversion.
80	4	48	Table Y Scaler	32-bit floating point value

84	4	52	Independent Variable Scaler	32-bit integer
84	2	56	Computed Value Mapping Table	Pairs of one byte values relating Measured Independent Variable to Calculated Value (in increasing order of independent variable). The first value is an independent variable the second value is the computed value. Last enty is denoted by 0xFF in both fields. If that is the only entry, the Computed Value is equal to the Independent Value.
				End of ***Sensor Block***
65408	128		Debug	*The end of the address map is reserved as a scratch space for Debug / Development*