MICROCHIP

AN3403

Indoor Air Quality Monitor: Concept and Implementation

Introduction

Authors: Ankit Phougat, Namrata Dalvi, Microchip Technology Inc.

Air pollution refers to a wide range of gases, liquids and solids suspended in the atmosphere, which have harmful effects on human health. Since most people spend a significant amount of their time indoors, the indoor air pollution and environment have a determining influence on human health. The indoor air pollution is one of the world's largest environmental problems. Based on figures from the World Health Organization, 3.8 million people die prematurely each year from illness attributable to household air pollution. Hence, it is essential to have good air quality monitor systems at the office and at home for constant air quality monitoring.

The indoor air has major airborne contaminants, such as total volatile organic compounds (TVOC), Carbon dioxide (CO_2) and Particulate Matter with a dust particle diameter of up to 2.5 microns ($PM_{2.5}$). The indoor air has two other factors that affect its quality: The humidity and temperature levels of the indoor environment. The Indoor Air Quality Monitor (AQM) is a real-time monitoring system, which measures the values of these contaminants and factors. The smart and secure AQMs are capable of transmitting the monitored air quality parameters to a cloud server through the wireless communication medium for real-time data modeling, real-time pollution data mapping, smart notification, and to generate automated reports.

In this application demonstrator, the AQM system is designed and implemented using Core Independent Peripherals (CIPs) and intelligent analog peripherals of the ATmega4808 microcontroller (AVR® MCU) featuring the 8-bit AVR processor. Microchip's CryptoAuthentication™ secure element (ATECC608A) and a fully certified Wi-Fi® module (ATWINC1510) are used to securely connect the smart AQM to the Google Cloud™ IoT Core platform.

The application demo is realized using the AVR-IoT WG development board from Microchip, and sensors and several click boards from MikroElektronika. Refer to the AVR-IoT WG Development Board product page for AVR-IoT WG development board details.

AQM Documentation Overview

The following documents cover the AQM documentation:

1. This application note describes the hardware and firmware overview of the AQM system and its power considerations.
2. The supplemented firmware creation guide *"AN3417 - Indoor Air Quality Monitor: Firmware Creation Using Atmel START and MPLAB® Code Configurator (MCC)"* covers the usage of AVR-IoT stack for AQM application. Additionally, it covers configuration details of microcontroller peripherals and click boards, using Atmel START and MCC.
3. The *"Indoor Air Quality Monitor: User Guide"* covers hardware setup, hardware connections, operating steps, LED indications, provisioning of the AVR-IoT WG board's Wi-Fi module, and visualization of AQM Data over the cloud.

Overview of Outdoor and Indoor Air Quality

The outdoor air pollution is well known and is mainly due to industrial emissions, auto exhaust, soot and other factors.

The indoor air quality is a major concern in offices and houses due to the significant impact of air pollution on health, comfort, well-being, and productivity of the occupants. Scientific studies show that many premature deaths are reported each year around the world due to indoor air pollution. According to the World Health Organization (WHO), these deaths are attributable to pneumonia, stroke, ischemic heart disease, chronic obstructive pulmonary disease (COPD), and lung cancer.

Most of the outdoor air quality monitoring systems or stations are established by the national and state government bodies. These outdoor air quality sensors or monitors are mostly designed to detect toxins such as ozone, vehicle emissions, particle pollution, Nitrogen dioxide (NO_2), Sulfur dioxide (SO_2) and Carbon monoxide (CO).

The indoor air quality monitors, on the other hand, are more often designed to detect particle pollution ($PM_{2.5}$, PM_{10}), volatile organic compounds (VOC) and Carbon dioxide (CO_2).

$PM_{2.5}$ particulate matter with particle diameter up to 2.5 microns is among the most dangerous air pollutants. Due to their small size, $PM_{2.5}$ particles can travel deep into the human lung and cause a variety of health issues; for instance, by triggering asthma attacks or contributing to cardiovascular disease. The TVOC is considered as an important indicator for indoor hygiene and indoor air quality. Elevated TVOC levels can cause headaches and irritation. Some TVOCs are carcinogens (e.g., formaldehyde), and some are irritants at normal levels (e.g., toluene). Hence, the measurement of TVOC levels is important in the case of an indoor air quality monitor. Higher Carbon dioxide levels in the indoor environment may cause the occupants to grow drowsy, to get headaches, or to function at lower activity levels. Indoor CO_2 levels are an indicator of the adequacy of outdoor air ventilation relative to indoor occupant density and metabolic activity. Hence, the CO_2 levels are measured by an indoor air quality monitor.

The quality of the air is decided by the Air Quality Index (AQI) value. National agencies have algorithms to calculate AQI based upon air quality parameters. Typically, a lower value of AQI indicates good air quality. The higher the AQI value, the greater the level of air pollution and health concern. For example, an AQI value of 50 represents good air quality with little potential to affect public health, while an AQI value over 300 represents hazardous air quality. An AQI value of 100 corresponds to the typical air quality standard for the pollutant, which is the level the Environmental Protection Agency (EPA) has set to protect public health. AQI values below 100 are generally thought of as satisfactory. For certain sensitive people groups, air quality is unhealthy if AQI values are above 100, and unhealthy for everyone if AQI is above 151. Refer to the Air Quality Index Guide by Environmental Protection Agency in 9. References for more details.

Table of Contents

Introduction......1

Overview of Outdoor and Indoor Air Quality......2

1. Indoor Air Quality Monitor Overview...... 4

2. Hardware Overview......6
   - 2.1. AVR-IoT WG Development Board......6
   - 2.2. MikroElektronika's Click Boards and Sensors......7
   - 2.3. Modification to mikroBUS™ Header for 5V Power Supply...... 9
   - 2.4. Features of ATmega4808 Used in the AQM Application 9

3. Firmware Overview...... 12
   - 3.1. Firmware Components...... 12
   - 3.2. AQM Application Firmware 14

4. Firmware Development...... 18
   - 4.1. Software Tools......18
   - 4.2. Firmware Configuration and Generation...... 18
   - 4.3. Programming the Microcontroller...... 18

5. Application Demo...... 19

6. Memory Requirement......20

7. Power Considerations 21
   - 7.1. Low Power Considerations......21

8. Conclusion...... 23

9. References......24

10. Appendix...... 25
    - 10.1. AQM Schematic...... 25
    - 10.2. Visualization of AQM Parameters...... 25

11. Revision History...... 27

The Microchip Website......28

Product Change Notification Service......28

Customer Support...... 28

Microchip Devices Code Protection Feature......28

Legal Notice...... 28

Trademarks...... 29

Quality Management System...... 29

Worldwide Sales and Service......30

1. Indoor Air Quality Monitor Overview

This application note discusses the AQM implemented using the AVR-IoT WG development board, and sensors and several click boards from MikroElektronika. The AVR-IoT development board is equipped with the Microchip's ATmega4808 microcontroller, CryptoAuthentication secure element (ATECC608A), and Wi-Fi module (ATWINC1510).

The scope of this application is limited to the development of the AQM system using available hardware modules and application firmware using the AVR-IoT WG development board's firmware (AVR-IoT stack). The AQM system uses available click boards from Mikroelektronika for sensors, EEPROM, and OLED display, except humidity and particulate matter (PM) sensors. The PM sensor and the humidity sensor are connected using a PROTO click. Hardware and firmware details of the AQM system are described in the respective sections of this document.

The AQM uses the Microchip sandbox account, which is registered with the Google Cloud IoT Core platform. The AVR-IoT WG development board is preconfigured to communicate with this account. A web application (i.e., webpage) is designed specifically for the AVR-IoT WG development board to visualize the data as graphs.

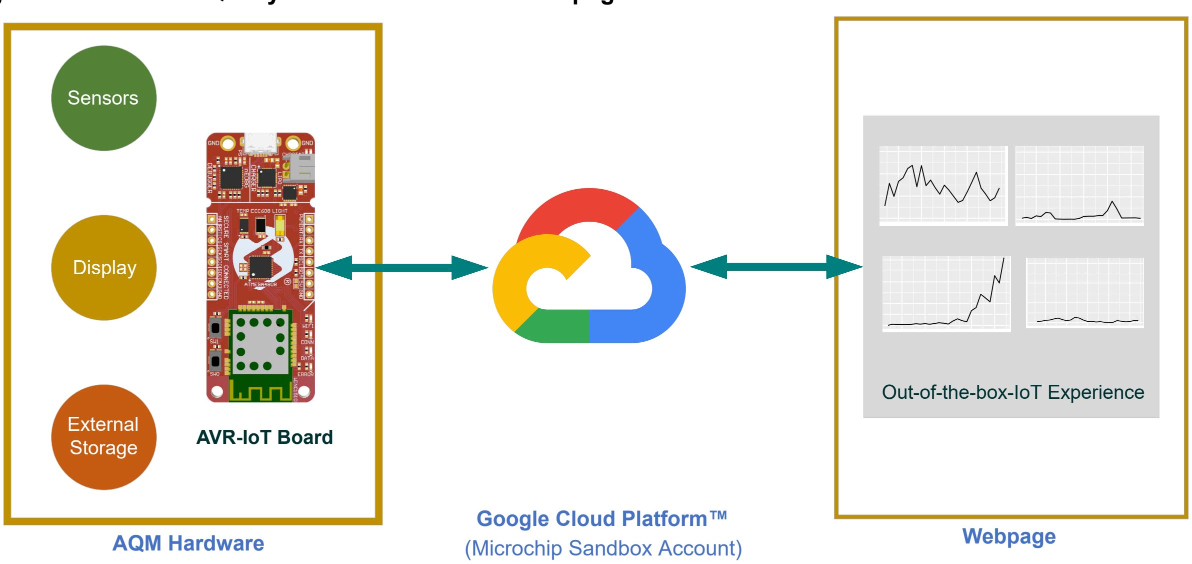
The AQM uses the designed webpage (avr-iot.com) to visualize air quality parameters.

Migrating to a private Google Cloud account and web application development is not in the scope of this application note. Refer to AVR-IoT Google Cloud Setup for more information.

After power-up, the AQM searches for the preconfigured Wi-Fi router to connect. If the Wi-Fi router is available, the AQM connects to it and gets access to the Internet. If the Wi-Fi router is not available, it continues to search and turns on the red LED to indicate Wi-Fi connection error. The microcontroller monitors indoor environmental temperature and humidity along with main airborne contaminants such as $PM_{2.5}$, CO_2 and TVOC. The microcontroller processes these acquired readings and calculates the AQI value from the readings of the $PM_{2.5}$ sensor. The AQI and other acquired air quality parameters are stored on external EEPROM and displayed on the OLED.

If the AQM finds Internet connectivity, it uploads the AQI and other parameters to Google Cloud. After that, the microcontroller enters Standby sleep mode, and wakes up periodically to monitor the sensors and transfer the processed data to the cloud. The microcontroller also wakes up from sleep instantly to a switch press event and displays the air quality parameters on the OLED. Figure 1-1 shows the overall AQM system with the cloud and webpage.

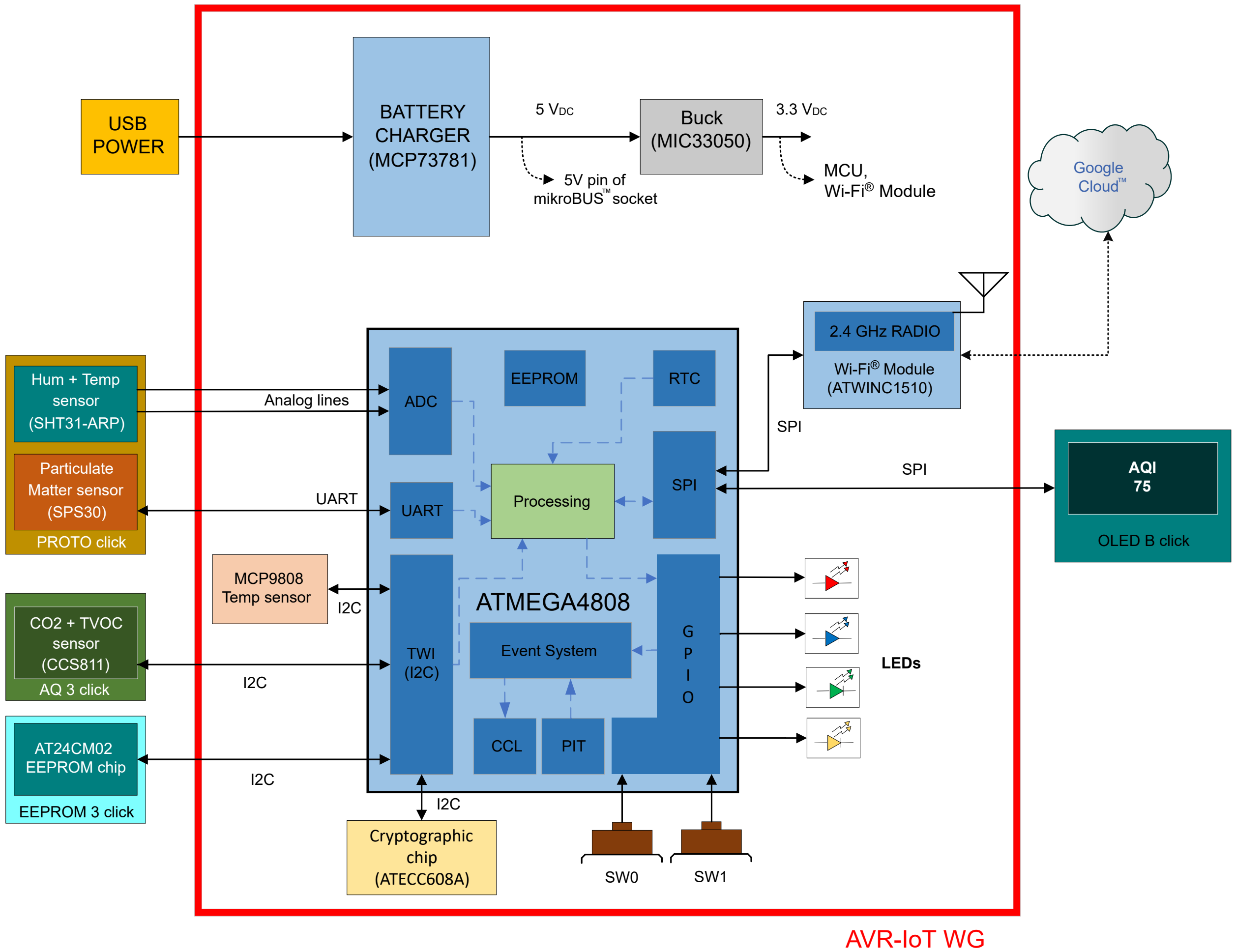
Figure 1-1. Overall AQM System with Cloud and Webpage



The communication between the microcontroller and Google Cloud is secured by the secure element ATECC608A, which is mounted on the AVR-IoT WG development board. The users can view all the air quality parameters by visiting a webpage designed for the out-of-the-box IoT experience with the AVR-IoT WG board. The webpage pulls data from Google Cloud every second and updates the air quality parameters graph.

The AQM system block diagram of the designed AQM is shown in Figure 1-2, which highlights the peripherals of the microcontroller used for this application and required interfaces to communicate with the external hardware.

Figure 1-2. AQM Block Diagram



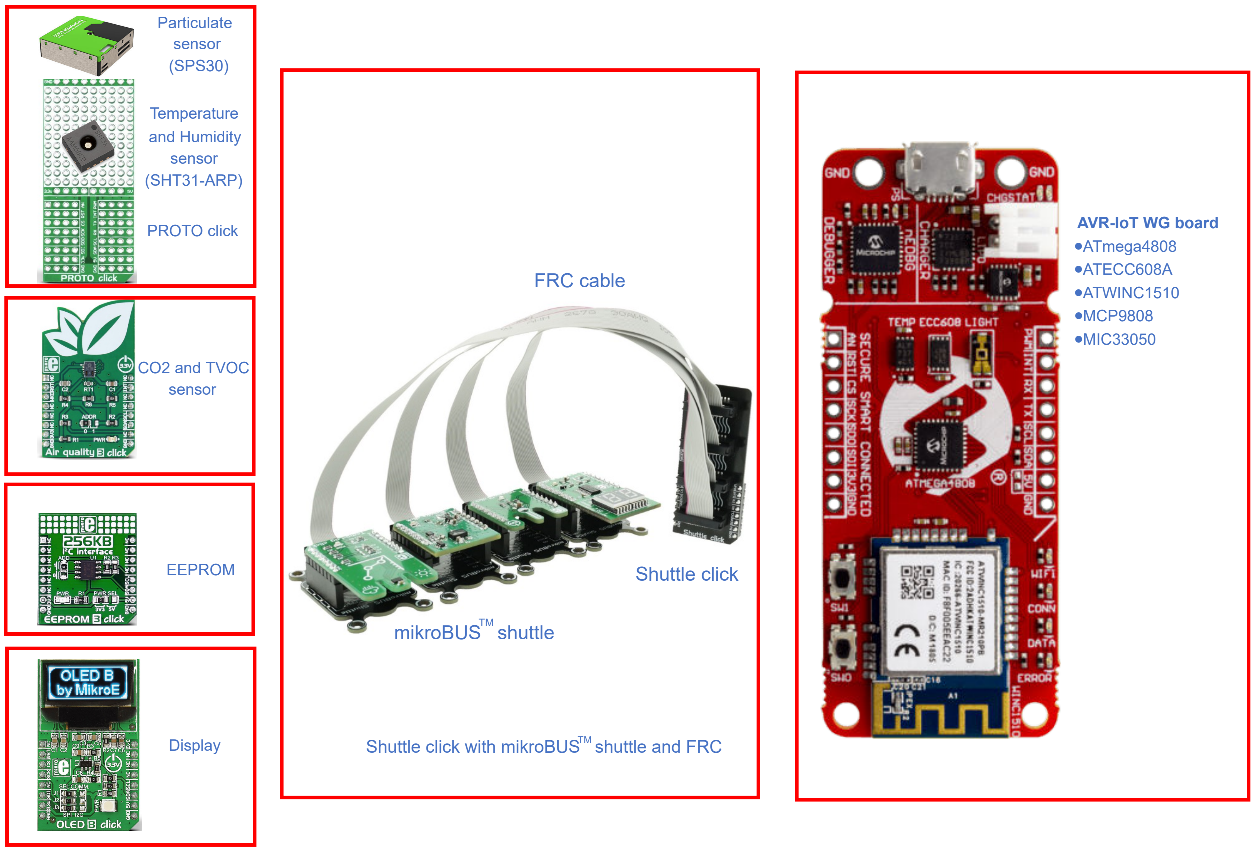
The designed AQM system offers the following features:

- Microchip megaAVR® microcontroller - ATmega4808 with CIPs and intelligent analog peripherals
- $PM_{2.5}$, CO_2, TVOC, humidity and temperature monitoring
- Hardware Security: CryptoAuthentication secure element ATECC608A
- Wireless connectivity: A fully certified Wi-Fi module ATWINC1510
- Google Cloud connectivity
- OLED display
- Data logging

2. Hardware Overview

The AQM hardware features an AVR-IoT WG development board, sensors mounted over MikroElektronika click boards, external EEPROM storage, and OLED display click board. The AVR-IoT WG development board has a single mikroBUS™ slot. A Shuttle click along with mikroBUS shuttles and Flat Ribbon Cables (FRC) are used to expand mikroBUS slots, as shown in Figure 2-1. Sensor clicks, external EEPROM click, and display click are placed over mikroBUS slots.

Figure 2-1. Click Boards Connection to AVR-IoT WG Development Board



Details of the hardware modules and their components, which are used for the implementation of the AQM system, are covered in the following sections.

2.1 AVR-IoT WG Development Board

The AVR-IoT WG development board combines a powerful 8-bit ATmega4808 microcontroller, an ATECC608A, and an ATWINC1510, which provide the most simple and effective way to connect the embedded application to the Google Cloud IoT Core platform. The board includes an on-board debugger and requires no external hardware to program and debug the microcontroller. For more details, refer to the AVR-IoT WG Development Board product page.

From the AVR-IoT WG development board, the AQM system uses a microcontroller (ATmega4808), a secure element IC (ATECC608A), a Wi-Fi module (ATWINC1510), a buck regulator (MIC33050), a digital temperature sensor (MCP9808), a mechanical switch and a mikroBUS connector. The following subsections provide detailed information about the main components of the AVR-IoT WG development board.

MCU-ATmega4808

The ATmega4808 is a microcontroller featuring the 8-bit AVR processor running at up to 20 MHz and with 48 KB Flash, 6 KB SRAM and 256 bytes of EEPROM. The microcontroller uses the latest CIPs with low-power features, including Event System, intelligent analog and advanced peripherals.

The peripherals of ATmega4808 used in the AQM system are:

- Analog-to-Digital Converter (ADC)
- Universal Synchronous and Asynchronous Receiver and Transmitter (USART)
- Serial Peripheral Interface (SPI)
- Dual-mode Master/Slave Two-Wire Interface (TWI): I^2C compatible
- Real-Time Counter (RTC)
- Periodic Interrupt Timer (PIT)
- Sleep Controller
- EEPROM Data Memory
- Event System
- Configurable Custom Logic (CCL)

Secure Element - ATECC608A
ATECC608A employs ultra-secure hardware-based cryptographic key storage and cryptographic countermeasures, which eliminate potential backdoors linked to software weaknesses. ATECC608A is used for storing the private and public key which are needed for secure IoT communication. ATECC608A communicates with a host controller over the I^2C interface.

Wireless Connnectivity - ATWINC1510 Wi-Fi® Module
ATWINC1510 is an IEEE 802.11 b/g/n IoT network controller. The ATWINC1510 Wi-Fi module integrates ATWINC1510 SoC, a 26 MHz oscillator, balun, and a printed antenna or a micro co-ax (μFL) connector for an external antenna. This module communicates with a host controller over the SPI interface.

Digital Temperature Sensor - MCP9808
MCP9808 digital temperature sensor serves a broad range of applications with an accuracy of 0.5 degrees from -20°C to +100°C, along with a high-temperature resolution of 12 bits. Additional features include shutdown, an under/over temperature monitor, and a critical-temperature alert. This digital sensor is connected to the microcontroller over an I^2C interface.

Buck Regulator - MIC33050
MIC33050 is a high-efficiency 600 mA PWM synchronous buck (step-down) regulator with an internal inductor. Its main features are input voltage range from 2.7V to 5.5V, 600 mA output current, 20 μA quiescent current, and Low Output Voltage Ripple.

2.2 MikroElektronika's Click Boards and Sensors

The AQM uses click boards from MikroElektronika as external hardware.

Shuttle Click, mikroBUS™ Shuttle and FRCs
The Shuttle click is a mikroBUS socket expansion board, which provides an easy solution for stacking up to four click boards on a single mikroBUS. Along with the Shuttle click, the mikroBUS shuttle with FRC is used for 1x4 expansion. In the AQM, the Shuttle click is placed over the AVR-IoT WG development board's mikroBUS slot, and four mikroBUS shuttles are connected with the Shuttle click using FRC cables. With this combination, four different clicks can be placed over mikroBUS shuttles.

PROTO Click for Humidity and PM Sensor Interface
The PROTO click is an accessory board with a mikroBUS form factor. It features a 10x11 prototyping area and additional power pads. It can be used to assemble custom electronics, creating a custom click board. The PROTO click is used to interface analog output humidity sensor and PM sensor with the microcontroller, as these sensor clicks are not available.

PM Sensor Module

The SPS30 sensor from Sensirion is an optical particulate matter sensor for air quality monitoring and control. The AQM monitors $PM_{2.5}$ using this sensor module. This sensor has a UART interface for communication with the microcontroller.

Humidity Sensor

SHT31-ARP is a fully calibrated, linearized, and temperature compensated analog output humidity and temperature sensor. It is connected to the ADC peripheral of the microcontroller. The AQM monitors humidity using this sensor module.

Note: For temperature measurement, the AQM uses an MCP9808 digital sensor mounted over the AVR-IoT WG development board.

The package of the SHT31-ARP sensor IC is a DFN package. This IC is mounted on a PROTO click using a DFN-8 to DIP-8 adapter. The pin connections between the DFN-8 to DIP-8 adapter and the PROTO click's mikroBUS connector are shown in Figure 2-2. The PM sensor (SPS30) is a separate module and has a 5-pin connector. The pin connections between SPS30 and PROTO click's mikroBUS connector are shown in Figure 2-2. Refer to Table 2-1 for more details.

Figure 2-2. PROTO Click mikroBUS™ Connector Connection with Humidity and PM Sensor

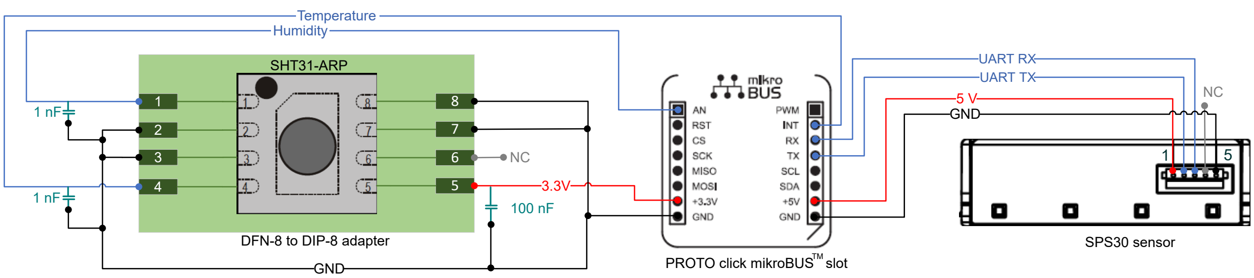


Table 2-1 shows the used humidity and PM sensor pins connected to the mikroBUS connector of PROTO click.

Table 2-1. mikroBUS™ PIN Connector with SHT31-ARP and SPS30 Sensors

SHT31-ARP	mikroBUS™ Connector of PROTO Click		SHT31-ARP	SPS30
PIN 1 (Humidity)	AN	PWM		
	RST	INT	PIN 4 (Temperature)	
	CS	RX		PIN 3 (**TX**)
	SCK	TX		PIN 2 (**RX**)
	MISO	SCL		
	MOSI	SDA		
PIN 5 (V_{DD})	3.3V	5V		PIN 1 (5V)
PIN 2,3,7,8 (V_{SS})	GND	GND		PIN 5 (GND)

Air Quality 3 Click

The AQM uses an Air quality 3 click for the TVOC and equivalent CO_2 (eCO2) measurement. Carbon dioxide equivalent is a measure used to compare the emissions from various greenhouse gases based upon their global warming potential. Air quality 3 click is equipped with the CCS811 that is an ultra-low power digital gas sensor solution that integrates a metal oxide (MOX) gas sensor to detect a wide range of volatile organic compounds (VOCs) and eCO2. This click is connected to the microcontroller over the I^2C interface.

OLED B Click

OLED B click carries a 96x39 px blue monochrome passive matrix OLED display. To drive the display, it has an SSD1306 controller. OLED B click communicates with the microcontroller through SPI.

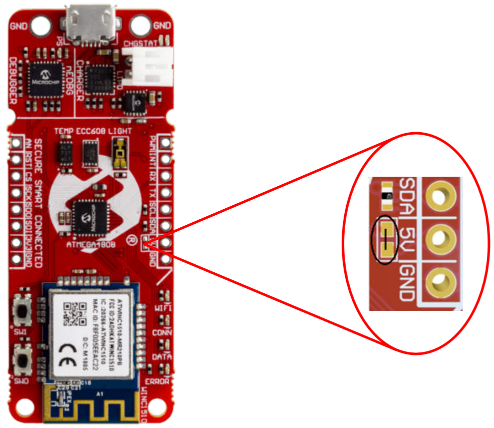
EEPROM 3 Click

EEPROM 3 click provides 256 KB of Electrically Erasable Programmable Read-Only Memory. The EEPROM module used is the AT24CM02, an I^2C serial EEPROM. One of the key features of the AT24CM02 is the Error Detection and Correction scheme (EDC), which allows error correction by utilizing six additional bits, internally assigned to a group of four bytes. This protection scheme can correct some types of bit errors, staying transparent to the end-user. The bit comparison and error correction are done internally. This click is connected to the microcontroller over the I^2C interface.

2.3 Modification to mikroBUS™ Header for 5V Power Supply

The SPS30 sensor requires 5 V_{DC} for operation. To use the SPS30 sensor with microBUS header, hardware modifications are needed. By default, the 5V supply is not connected to the mikroBUS header. To enable 5V to the header, short the footprint shown in Figure 2-3 by soldering the 0 ohm resistor.

Figure 2-3. Short 5V Jumper on AVR-IoT WG Board



Note: Refer to the Appendix section for AQM schematic and connections.

2.4 Features of ATmega4808 Used in the AQM Application

In this section, the microcontroller features and peripheral configuration details used in the AQM application are described.

ADC

The ADC of ATmega4808 can go up to a maximum conversion rate of 150 ksps with 8-bit resolution, or 115 ksps with 10-bit resolution. It has a built-in noise filtering capability. The noise countermeasures supported by the ADC are:

- Hardware sample accumulator
- Sampling delay
- Automatic sampling delay variation (ASDV)

In the AQM, ADC is configured with a 10-bit resolution and hardware sample accumulator with eight samples (fixed delay between samples). The ADC samples the voltage output of the Humidity sensor (SHT31-ARP) against the ADC voltage reference of V_{DD} (microcontroller input power supply).

USART

The USART peripheral is used to communicate with the PM sensor (SPS30). USART1 is configured with 115200 baud rate, Parity bit '0', and Stop bit '1' for interfacing with the $PM_{2.5}$ sensor.

SPI

The AQM uses SPI to communicate with the ATWINC1510 Wi-Fi device and display (OLED B click). The SPI data clock frequency is 2.5 MHz, and it is operating in Mode 0.

TWI (I^2C)
The AQM uses an I^2C compatible TWI to communicate with an ATECC608A cryptographic chip, external storage (EEPROM 3 click), digital temperature sensor (MCP9808), TVOC and CO_2 sensor (air quality 3 click board). TWI is configured with 100 kHz data clock frequency.

Sleep Controller
The ATmega4808 microcontroller supports three sleep modes (Idle, Standby and Power-Down). The AQM uses Standby sleep mode. In this mode, users can configure peripherals to run during the sleep period, using the respective RUNSTBY bit. In the AQM, CCL is enabled during the Standby sleep mode to detect switch press.

RTC
The AQM uses the RTC module to generate a periodic interrupt of 1 ms during the active period of the CPU. The 1 ms period is used by the firmware for scheduling the tasks. Refer 3. Firmware Overview for details.

PIT
This is a part of the RTC peripheral. It can be enabled independently and uses the same clock source as the RTC. In the AQM, PIT is configured to generate a periodic interrupt of two seconds using the RTC clock. PIT runs in all sleep modes. Hence, it is used in AQM to wake up the microcontroller from Standby sleep mode periodically.

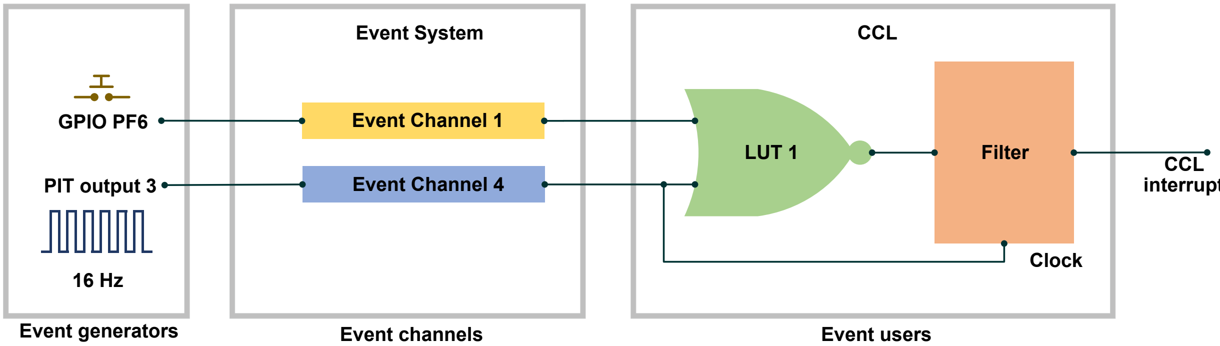
Event System
The Event System is a system for direct peripheral-to-peripheral signaling. The AQM uses two channels of the Event System. GPIO (PF6) is connected to input 1 of the LUT1 of CCL through Event Channel 1. PIT output 3 is connected to input 2 of the LUT1 of CCL through Event Channel 4. With this configuration, the Event System is used for the debouncing of switch SW0 input signal (mounted over AVR-IoT WG development board). Refer to the Switch Debouncing Implementation Using Event System, CCL and PIT section for more details.

CCL
The CCL is a programmable logic peripheral that can be connected to the device pins, to events, or other internal peripherals. The CCL peripheral is used in AQM to handle switch debouncing with the help of the Event System and PIT. Design and implementation details of switch debouncing are detailed in the Switch Debouncing Implementation Using Event System, CCL and PIT section.

EEPROM Data Memory
ATmega4808 features 256 bytes of EEPROM. The AQM uses the microcontroller's internal EEPROM for storing parameters (default configuration parameters, external EEPROM data storage index).

Switch Debouncing Implementation Using Event System, CCL and PIT
Switch debounce is implemented using Event System, CCL and PIT clock without any software intervention. The signal from a mechanical button often transitions several times between high and low each time the button is pushed or released, referred to as bounce. To use this signal and generate an event only once the button is pressed or released, some form of filtering needs to be implemented either in hardware or software, also referred to as debouncing. Debouncing the switch signal is accomplished by filtering it with the CCL, as shown in Figure 2-4, and using the filtered signal to generate an interrupt for the switch press event.

Figure 2-4. Switch Debouncing Using CCL Filter

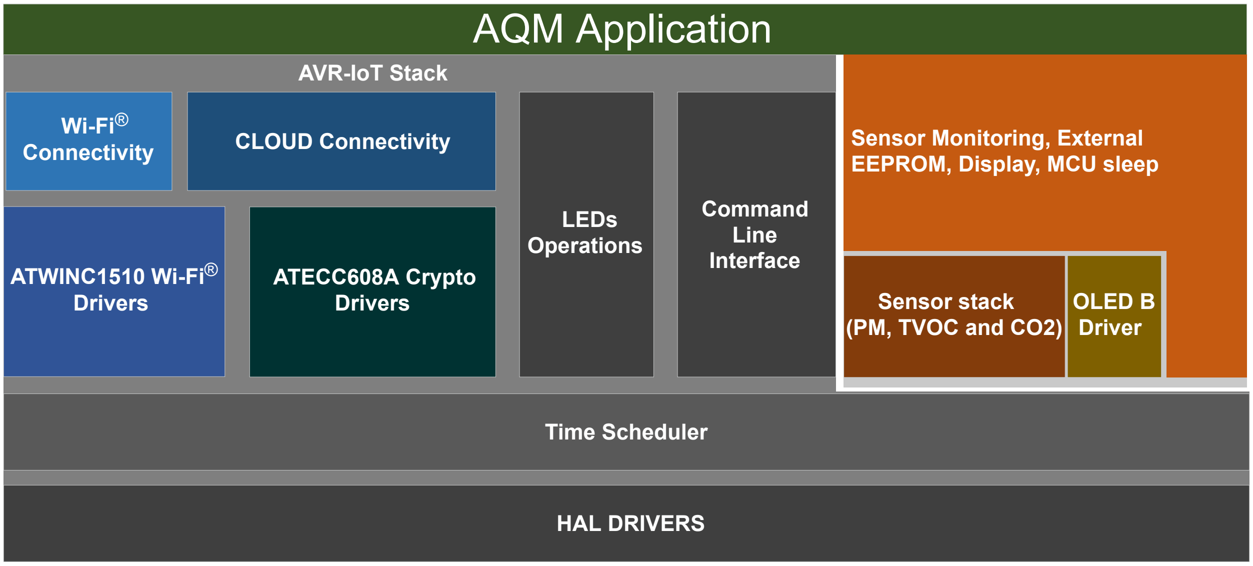


The filter requires a suitable clock frequency to operate. Any output from the LUT shorter than two clock cycles will be filtered out. In this application, PIT output 3 is used as the clock source (frequency: 16 Hz). The signals are routed via the Event System. Event Channel 1 is used for routing the switch signal, and Event Channel 4 is used for routing clock to event inputs of the LUT. Therefore, GPIO and PIT act as event generators and CCL acts as event user. For more details, refer to Getting Started with Core Independent Peripherals on AVR.

3. Firmware Overview

This section explains firmware implementation details of the indoor air quality monitor. The AQM application firmware is developed over the AVR-IoT stack; refer to the AVR-IoT WG Development Board and its documentation for more details about the AVR-IoT stack. Figure 3-1 shows the AVR-IoT stack and application firmware modules.

Figure 3-1. Application Firmware Layers



3.1 Firmware Components

3.1.1 AVR-IoT Stack

The AVR-IoT stack is a firmware library generated with the help of the Atmel START and MCC configurator tools. It is customized and configured for the out-of-the-box IoT experience using the AVR-IoT WG development board.

The AVR-IoT stack uses a custom time scheduler with the scheduling tick of 1 ms. RTC is used to generate the scheduling tick. Users can add or remove tasks to the scheduler queue along with the task's scheduling frequency. The Wi-Fi connection, Secure Cloud Connection, and LED indications are part of the AVR-IoT stack.

The communication with the Wi-Fi module is handled by the Wi-Fi task. This task handles initializing the Wi-Fi module, provisioning, connection to a Wi-Fi router, and data transmission.

Secure communication with Google Cloud is handled by the Cloud task. This task updates the cloud connection status. It tries to reconnect if the connection is lost.

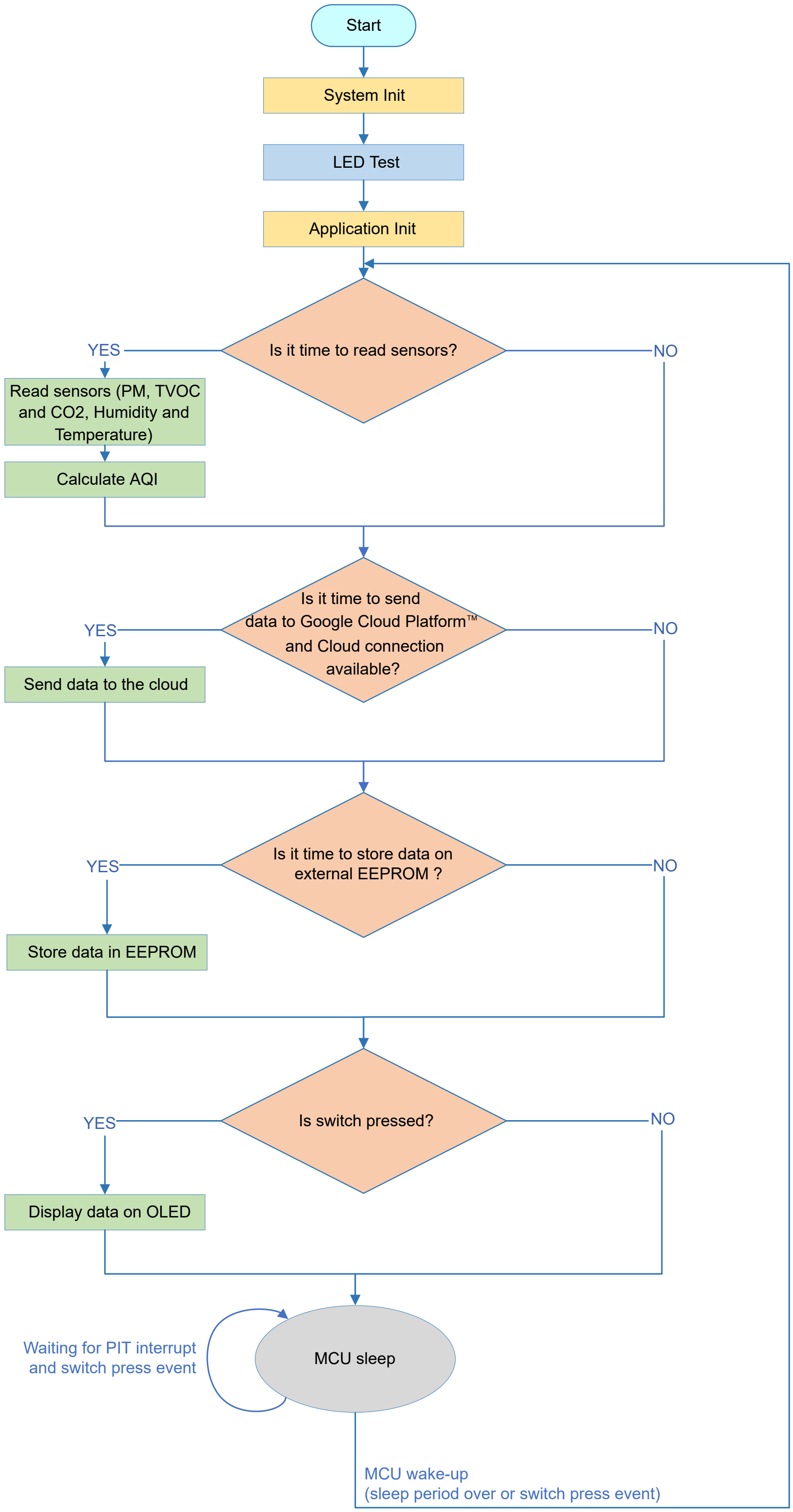
The AQM firmware uses the following AVR-IoT stack versions:

- AVR-IoT Stack for Atmel START: 1.1.1
- AVR-IoT Stack for MCC: 1.2.0

3.1.2 AQM Application Code

The application firmware periodically acquires data from the sensors and processes the acquired data. It calculates AQI and sends the processed data to Google Cloud. Additionally, the processed data are displayed on the OLED display and stored on external EEPROM for data logging. Figure 3-2 shows the flowchart of the application overview.

Figure 3-2. Firmware Flowchart



3.2 AQM Application Firmware

3.2.1 Initialization

The firmware execution starts with the initialization of the microcontroller and its peripherals. The AQM application uses microcontroller peripherals such as SPI, I^2C, UART, ADC, Digital I/O, Event System, CCLs, RTC and PIT. In AQM, the sleep functionality is also enabled. The system is initialized in the following configurations:

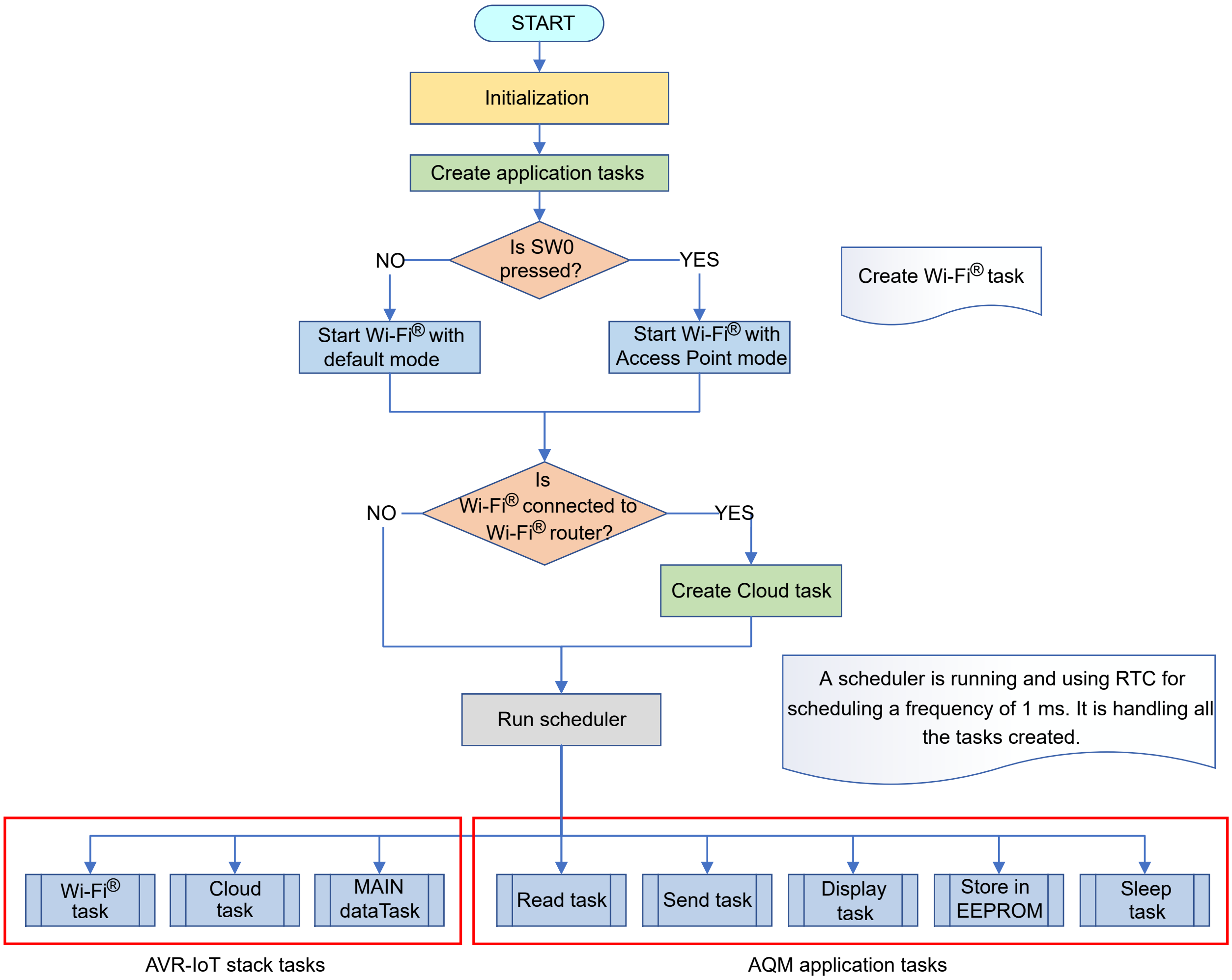
- SPI: SPI Data Clock (2.5 MHz), SPI Mode (MODE0)
- I^2C: I^2C Data Clock (100 kHz)
- USART1: Baud rate (115200), Parity bit (`0`), Stop bit (`1`)
- USART2: Baud rate (9600), Parity bit (`0`), Stop bit (`1`)
- ADC: Resolution (10-bit), sampling frequency (48 kHz) and eight samples accumulation with a fixed delay
- RTC is configured to generate an interrupt every 1 ms, and PIT is configured to generate an interrupt every two seconds.
- Event System and LUT1 of the CCL with filters are used for switch debounce.
- GPIOs as output for LEDs and SPI Slave Select (SS)
- GPIOs as input for SW0 and SW1
- Sleep Controller: Standby sleep mode is enabled. The CCL peripheral is configured to run in Standby sleep mode, and the CCL interrupt wakes up the microcontroller from sleep. The PIT also runs in Standby sleep mode, and the PIT interrupt also wakes up the microcontroller from sleep.

The AQM firmware is developed over the AVR-IoT stack, and after peripheral initialization, application tasks are created. The firmware execution (task scheduling) happens during the active period of the CPU. Following are the AQM application firmware tasks:

- `READ_Task()`: Handles reading values from sensors and AQI calculation.
- `DISPLAY_Task()`: Handles displaying the AQM parameters over OLED B click.
- `SEND_Task()`: Handles uploading the AQM parameters to the Google Cloud.
- `STORE_Task()`: Handles storing the AQM parameters on external EEPROM.
- `SLEEP_Task()`: Handles microcontroller's sleep and wake-up.

Figure 3-3 shows the flowchart of the application initialization.

Figure 3-3. Application Initialization



3.2.2 Application Tasks describes the AQM application firmware tasks along with their responsibilities.

3.2.2 Application Tasks

Read Sensors

This task handles the initialization of the PM sensor and AQ3 click. It reads all sensors (PM, CO_2, TVOC, Temperature and Humidity) data and calculates the AQI based upon $PM_{2.5}$ readings. The reading of the sensors is done periodically (once every four seconds).

The interfacing details of the sensors with the microcontroller and their data acquisition information are as follows:

Temperature Sensor

The microcontroller is interfaced with a digital temperature sensor (MCP9808) over I^2C. The function `SENSORS_getTempValue()` reads the temperature value from the MCP9808 sensor (the I^2C address of MCP9808 is 0x18).

Humidity Sensor

The humidity sensor (SHT31-ARP) has an analog voltage output, hence it is interfaced with the microcontroller through the ADC peripheral. The ADC peripheral is configured to sample the output of the humidity sensor at 48 kHz. The ADC filters the sampled signal by accumulating eight samples using a hardware sample accumulator with a fixed delay. The function `ADC0_GetConversion()` gets the humidity value from the sensor (SHT31-ARP).

TVOC and eCO2 Sensor

The Air quality 3 click has a CCS811 sensor, and measures TVOC and equivalent Carbon dioxide reading (eCO2). It communicates with the host microcontroller over the I^2C interface. The function `ccs811_GetResults()` reads the TVOC and CO_2 from the Air quality 3 click (I^2C address of Air quality 3 click is 0x5A).

Note: As per the CCS811 sensor data sheet, it is recommended that the sensor runs for 48 hours for the first time to burn-in, and then 20 minutes in the desired mode every time the sensor is used. During these 20 minutes, the user may observe a constant value as 0 for TVOC and 400 for eCO2. In the application firmware, 48 hours of burn-in time and 20 minutes delay are not addressed. In the AQM, TVOC, and CO_2, the readings may not be accurate until 20 minutes after a power cycle.

$PM_{2.5}$ Sensor

The microcontroller communicates with the PM sensor (SPS30) module over the UART interface. Data from the SPS30 sensor contains values of PM_{10}, $PM_{0.5}$, $PM_{1.5}$, and $PM_{2.5}$ (PM stands for particulate matter; the number describes the size of dust in microns). The indoor AQM only uses the $PM_{2.5}$ data. The function `sps30_read_measurement ()` reads the $PM_{2.5}$ value from the PM sensor (SPS30).

After reading $PM_{2.5}$ values, the firmware calculates the AQI based on the $PM_{2.5}$ reading. The AQI value is an index for reporting air quality. The AQI value of the surrounding air less than 50 is considered good air quality.

This application uses the AQI calculation formula provided by the United States Environmental Protection Agency.

AQI Calculation Formula:

$$I = \frac{I_{high} - I_{low}}{C_{high} - C_{low}}\left(C - C_{low}\right) + I_{low}$$

Where:

I = the AQI

C = the pollutant concentration

C_{low} = the concentration breakpoint that is ≤ C

C_{high} = the concentration breakpoint that is ≥ C

I_{low} = the index breakpoint corresponding to C_{low}

I_{high} = the index breakpoint corresponding to C_{high}

Find more details about the AQI at the Air Quality Index (AQI) - EPA

$PM_{2.5}$ Average Calculation:

The AQI calculation requires hourly data of the PM sensor for the past 12 hours. The AQM application firmware implements a method provided by AirNow to take an average of the past 12-hour values.

Average PM Calculation Method

1. Compute the $PM_{2.5}$ concentrations range (max. and min.) over the last 12 hours.
2. Divide the range by the maximum concentration in the 12-hour period to obtain the scaled rate of change.
3. Compute the weight factor by subtracting the scaled rate from 1. The weight factor must be between 0.5 and 1. The minimum limit approximates a 3-hour average. If the weight factor is less than 0.5, then limit the value to 0.5.
4. Multiply each hourly concentration by the weight factor raised to the power of how many hours ago the concentration was measured (for the current hour, the factor is raised to the zero power).
5. Compute the average PM by summing these products and divide by the sum of the weight factors raised to the power of how many hours ago the concentration was measured.

Note: This demo does not take hourly sensor readings. For a demonstration of real-time data, all the sensors are being measured every four seconds. The users can configure the sensor measurement period to one hour.

Display Air Quality Parameters

This task displays the air quality parameters when the switch SW0 of the AVR-IoT WG development board is pressed. After the switch SW0 press event, the air quality parameters sequentially display on the OLED. Each air quality parameter remains on the OLED screen for approximately four seconds, and then the next parameter is displayed. Press switch SW0 to instantly change the air quality parameter on the OLED. After displaying all the AQM parameters, the OLED display is turned off. If the switch is pressed again, the OLED turns on and starts displaying the air quality parameters.

Send Data to Cloud

This task uploads the AQM data over Google Cloud. The uploading of the AQM data is done periodically (once every four seconds) if the cloud connection is available. The AVR-IoT stack handles the connection between the microcontroller and Google Cloud, and this task handles uploading the parameters to Google Cloud. The AQM application firmware uses the Message Queuing Telemetry Transport (MQTT) protocol to publish (upload) data to Google Cloud. The data are transferred from the sensor to the cloud through a JSON object: An ASCII string formatted as follows: { 'AQI': XXX, 'Temp': YYY, 'RH': XXX, 'PM': YYY,'TVOC': XXX, 'CO2': YYY }, where XXX and YYY are numerical values expressed in decimal notation.

Note:

1. On the AVR-IoT WG development board webpage, the graphs are constructed by using the name of the parameter and its value.
2. JSON is a concise hierarchical data serialization syntax supported by all modern browsers. Its format makes it a lightweight way of representing objects while remaining human-readable.

The application firmware is storing data on external EEPROM. This task also checks if the AQM has old data to be sent (the data which were stored on external EEPROM during no cloud connection). It uploads old data first then it uploads the real-time data.

Data Storage to External EEPROM

This task stores the AQM data on external storage (EEPROM 3 click). It accesses the external storage with the help of the I^2C interface and stores the air quality parameters (AQI, $PM_{2.5}$, CO_2, TVOC, RH, and temperature) on external storage. It stores the AQM data periodically (i.e., for every one minute).

MCU Sleep

This task is responsible for the sleeping cycle of the microcontroller. After the completion of all the tasks, this task calls the `sleep()` command and puts the microcontroller into a configured Standby sleep mode. It switches the main oscillator of the microcontroller from 16/20 MHz (OSC20M) to 32.768 kHz Ultra-Low Power (OSC32K) before calling the `sleep()` command. It switches the main oscillator back to OSC20M when the microcontroller wakes up from sleep.

In the AQM, the microcontroller is using two wake-up sources: PIT interrupt and CCL interrupt. The PIT interrupt wakes up the microcontroller from sleep periodically (once in every two seconds). The switch SW0 press event can also wake up the microcontroller from sleep.

4. Firmware Development

4.1 Software Tools

Atmel Studio and MPLAB X IDEs are used for the firmware development.

The toolchain to develop firmware using Atmel Studio:

- Atmel Studio 7 (v7.0.2397)
- AVR GCC Compiler (v5.4.0)
- Atmel START Configurator Tool (v1.7.279)

The toolchain to develop firmware using MPLAB X IDE:

- MPLAB X IDE (v5.35)
- AVR GCC Compiler (v5.4.0)
- MPLAB Code Configurator (MCC) (v3.95.0)

Note: To run the demo, the installed tool versions must be the same or later. This example is not tested with the previous versions.

4.2 Firmware Configuration and Generation

The firmware is available on GitHub, as below, and in the Atmel START examples.

View Code Example on GitHub (Atmel Studio)
Click to browse repository

View Code Example on GitHub (MPLAB® X)
Click to browse repository

The application firmware can be generated using Atmel START or MCC.

For more details about the process to generate firmware using Atmel START or MCC frameworks, refer to *"AN3417 - Indoor Air Quality Monitor: Firmware Creation Using Atmel START and MPLAB® Code Configurator (MCC)"*.

4.3 Programming the Microcontroller

To program application firmware on the microcontroller, refer to AVR-IoT WG Development Board User Guide.

5. Application Demo

Refer to the *"Indoor Air Quality Monitor: User Guide"* for hardware setup, hardware connections, operating steps, LED indications, provisioning of the AVR-IoT WG board's Wi-Fi module, and visualization of AQM Data over the cloud.

Refer to the Appendix section for the AQM schematic and visualization of AQM parameters on the webpage.

6. Memory Requirement

The application firmware program and data memory requirements with Atmel Studio and MPLAB X IDEs are as given below.

Table 6-1. Memory Requirements with Atmel Studio 7 (v7.0.2397)

Optimization	Program Memory	Data Memory
-Os	44416 (90.4%)	2781 (45.3%)

Table 6-2. Memory Requirements with MPLAB® X IDE (v5.35)

Optimization	Program Memory	Data Memory
-Os	47272 (96.2%)	2771 (45.1%)

7. Power Considerations

The AVR-IoT board is powered by a USB port (5 V_{DC}). The board is equipped with a buck converter MIC33050 for generating 3.3V, which is used as V_{DD} for the microcontroller and other external hardware except for the PM sensor. The PM sensor requires 5 V_{DC} supply, which is given by shorting a 5V jumper at AVR-IoT WG board.

The power consumption data of the AQM is captured while the microcontroller is operating in Active mode.

Note: The current is measured after the buck converter (3.3 V_{DC}).

Table 7-1. Total Current Consumption at 3.3 V_{DC} Supply in Active Mode

Components in AQM	Typical Current Consumption [mA]
External sensor hardware (Except the PM sensor)	29
AVR-IoT board	96
Total	**125**

Table 7-2. Total Current Consumption at 5 V_{DC} Jumper (PM Sensor only) in Active Mode

Components in AQM	Typical Current Consumption [mA]
SPS30 sensor module	50

Note: This application uses existing hardware click boards. Standby sleep mode is implemented for the microcontroller only. All other click boards and the Wi-Fi module are always ON. For demonstration purposes, the sensor data are uploaded to Google Cloud every four seconds. As per webpage specifications, the device sends data every four seconds.

Further power consumption can be reduced by turning off the Wi-Fi module and sensor clicks.

7.1 Low Power Considerations

The current consumption of the AQM is higher than the typical IoT sensor nodes, as it is built using available hardware modules (AVR-IoT WG development board and MikroElektronika's clicks). This current consumption can be reduced to a few µA if customized hardware is used to build the AQM system. This can be achieved by switching the microcontroller oscillator to ultra-low power, putting the microcontroller and the Wi-Fi module into Sleep state, and controlling the power supply of the sensors. The system power can be reduced further by decreasing the sensor readings, EEPROM writing, and data upload frequency.

Table 7-3. Current Consumption by the AQM when the Microcontroller and the Wi-Fi® Module in Sleep State

Components in AQM	Typical Current Consumption [mA]
AQM (microcontroller and Wi-Fi sleep) with Shuttle click and Sensor clicks connected (@ 3.3 V_{DC} supply)	~29 mA
SPS30 sensor module (@ 5 V_{DC} supply)	50 mA

Table 7-4. Current Consumption by the AQM (without External Hardware Connected) when the Microcontroller and the Wi-Fi® Module are in Sleep State

Components in AQM	Typical Current Consumption [mA]
AQM (microcontroller and Wi-Fi sleep) without Shuttle click connected (no external hardware clicks)	~200 µA

To measure the current consumption of the ATmega4808 microcontroller, the ATmega4809 Xplained Pro board is used as both Atmega4808 and ATmega4809 belong to the same family. The microcontroller is programmed with the AQM firmware. For current readings, the Data Visualizer tool with Power Debugger is used.

Table 7-5. Example of Current Consumption by the ATmega4808/9 Microcontroller in Standby Sleep Mode

Evolution Board	ATmega4809 Xplained Pro
Operating Voltage	3.3 V_{DC}
Sleep Mode	Standby sleep mode
Peripherals Running During Sleep	PIT, CLC
Current in Active mode (oscillator OSC20M), main clock operating at 10 MHz	4.8 mA
Current in sleep mode (oscillator OSC20M), main clock operating at 10 MHz	1.6 mA
Current in sleep mode (oscillator OSCULP32K), main clock operating at 32.768 kHz	~200 µA

Refer to the Low-Power Design Guide and AVR® Low-Power Techniques application notes for more information on how to achieve a design with low power consumption.

8. Conclusion

This application note covers the implementation details of the AQM solution using the AVR-IoT WG development board featuring ATmega4808, and sensors and several click boards from MikroElektronika. It also highlights the usage of CIPs like Event System, CCL, and intelligent analog peripherals of the ATmega4808 microcontroller for realizing the AQM system.

The embedded design can be secured with the ATECC608A CryptoAuthentication secure element, which handles the authentication of each device. Wireless connectivity can be seamlessly added to the design with a Wi-Fi module (ATWINC1510).

9. References

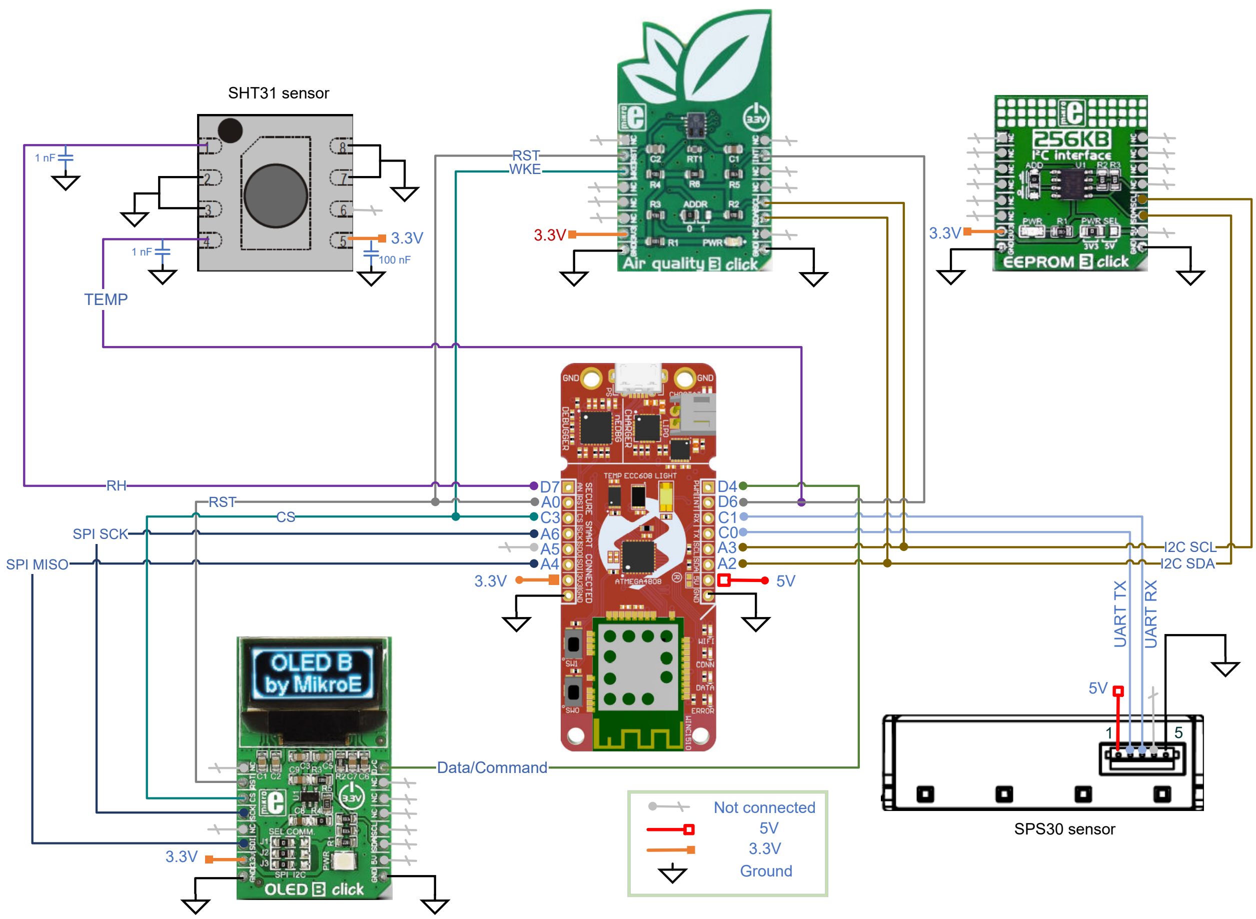
1. Air Quality Index - A Guide to Air Quality and Your Health Guide: https://www3.epa.gov/airnow/aqi_brochure_02_14.pdf
2. AQI Calculator: https://airnow.gov/index.cfm?action=airnow.calculator
3. ATmega4808 Product Page: https://www.microchip.com/wwwproducts/en/ATMEGA4808
4. ATmega4809 Product Page: https://www.microchip.com/wwwproducts/en/ATMEGA4809
5. Introduction to CIPs: http://ww1.microchip.com/downloads/en/AppNotes/Getting-Started-with-Peripherals-on-AVR-MCU-00002451C.pdf
6. Noise Countermeasures for ADC Applications: http://ww1.microchip.com/downloads/en/AppNotes/AN2551-Noise-Countermeasures-for-ADC-Applications-00002551C.pdf
7. Create Smart, Secure Nodes for Google Cloud IoT with the AVR-IoT WG Development Board: https://www.microchip.com/design-centers/internet-of-things/google-cloud-iot/avr-iot
8. AVR-IoT WG development board user guide: http://ww1.microchip.com/downloads/en/DeviceDoc/AVR-IoT-WG-Development-Board-User-Guide-50002809B.pdf
9. Household air pollution and health: https://www.who.int/news-room/fact-sheets/detail/household-air-pollution-and-health

10. Appendix

10.1 AQM Schematic

All the external clicks are connected to the single mikroBUS connector of the AVR-IoT WG development board. Figure 10-1 shows the AQM schematic details. The Shuttle click is not shown here.

Figure 10-1. AQM Schematic

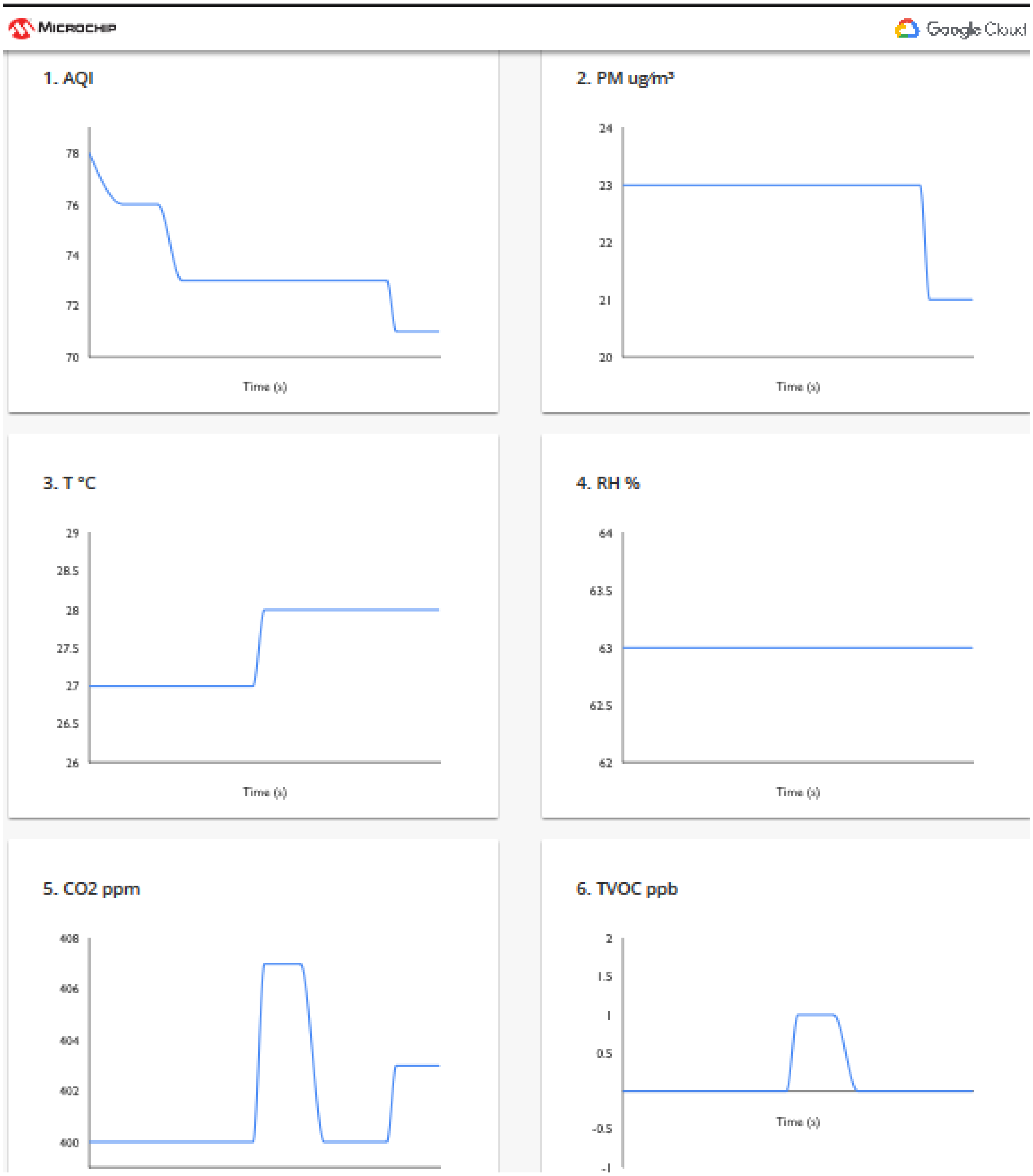


10.2 Visualization of AQM Parameters

Once the board is connected to Google Cloud through the Wi-Fi interface, the AQM starts uploading data to the cloud. The yellow LED flashes while the data are being uploaded. The avr-iot.com webpage shows the data gathered from various sensors in a graphical view, as shown in Figure 10-2.

Note: The order of the parameters shown in the figure can be changed by modifying the order in the JSON object in the firmware.

Figure 10-2. Graphical Display of Air Quality Parameters on Webpage



11. Revision History

Document Revision	Date	Comments
A	03/2020	Initial document release

The Microchip Website

Microchip provides online support via our website at http://www.microchip.com/. This website is used to make files and information easily available to customers. Some of the content available includes:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQs), technical support requests, online discussion groups, Microchip design partner program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Product Change Notification Service

Microchip's product change notification service helps keep customers current on Microchip products. Subscribers will receive email notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, go to http://www.microchip.com/pcn and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Embedded Solutions Engineer (ESE)
- Technical Support

Customers should contact their distributor, representative or ESE for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in this document.

Technical support is available through the website at: http://www.microchip.com/support

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with

Trademarks

ISBN: 978-1-5224-5666-7

Quality Management System

For information regarding Microchip's Quality Management Systems, please visit http://www.microchip.com/quality.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: http://www.microchip.com **Atlanta** Duluth, GA Tel: 678-957-9614 Fax: 678-957-1455 **Austin, TX** Tel: 512-257-3370 **Boston** Westborough, MA Tel: 774-760-0087 Fax: 774-760-0088 **Chicago** Itasca, IL Tel: 630-285-0071 Fax: 630-285-0075 **Dallas** Addison, TX Tel: 972-818-7423 Fax: 972-818-2924 **Detroit** Novi, MI Tel: 248-848-4000 **Houston, TX** Tel: 281-894-5983 **Indianapolis** Noblesville, IN Tel: 317-773-8323 Fax: 317-773-5453 Tel: 317-536-2380 **Los Angeles** Mission Viejo, CA Tel: 949-462-9523 Fax: 949-462-9608 Tel: 951-273-7800 **Raleigh, NC** Tel: 919-844-7510 **New York, NY** Tel: 631-435-6000 **San Jose, CA** Tel: 408-735-9110 Tel: 408-436-4270 **Canada - Toronto** Tel: 905-695-1980 Fax: 905-695-2078	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4485-5910 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-72400 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-72884388 **Poland - Warsaw** Tel: 48-22-3325737 **Romania - Bucharest** Tel: 40-21-407-87-50 **Spain - Madrid** Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 **Sweden - Gothenberg** Tel: 46-31-704-60-40 **Sweden - Stockholm** Tel: 46-8-5090-4654 **UK - Wokingham** Tel: 44-118-921-5800 Fax: 44-118-921-5820