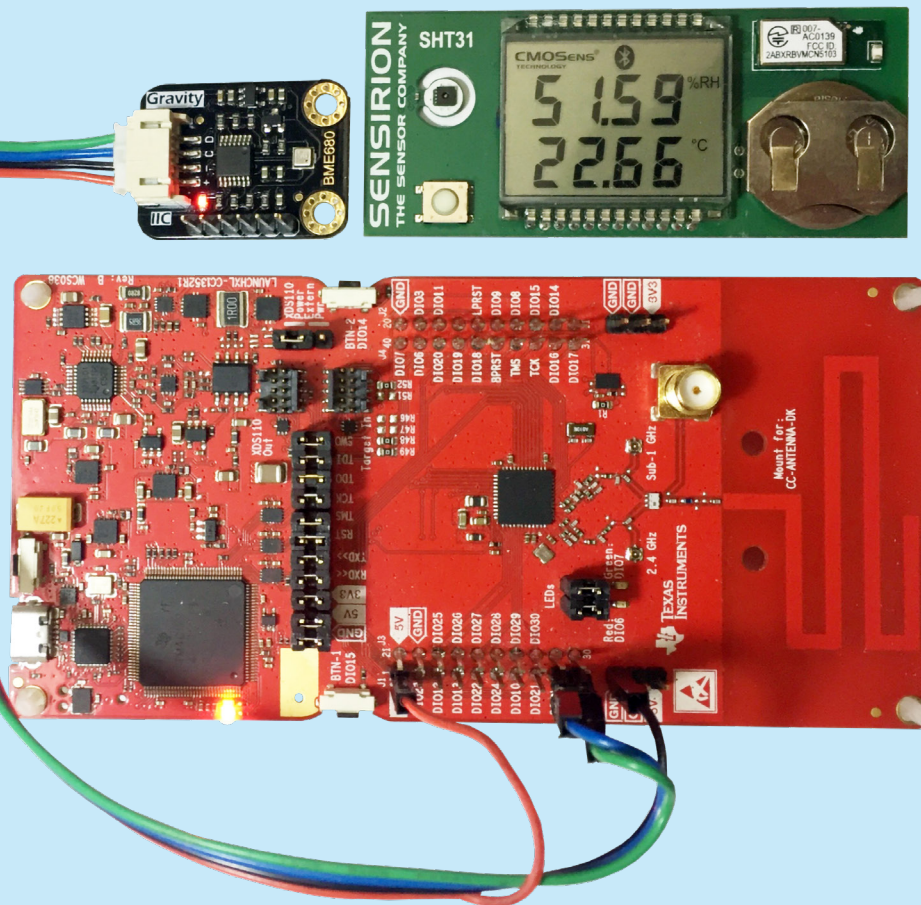


FTP WSZYSTKIE
POPRIEDNIE ODCINKI
KURSU SĄ DOSTĘPNE
NA STRONIE
WWW.MEDIA-AVT.PL

Ponowne podziękowania dla pana Marcina Paszkiewicza, szefa oddziału polskiego firmy Texas Instruments za udostępnienie wersji produkcyjnej zestawów startowych CC1352R1 LaunchPad.



Systemy dla Internetu Rzeczy (36)

Pomiar jakości powietrza, ciśnienia, wilgotności i temperatury czujnikiem BME680 dołączonym do zestawu startowego CC1352R1 LaunchPad

Czujnik jakości powietrza BME680 firmy Bosch jest bardzo popularny w zastosowaniach Internetu Rzeczy. Jest dostępny spory wybór pakietów programowych do jego użycia w środowisku Arduino. Jednak brak doniesień o dołączaniu tego czujnika do popularnego procesora komunikacyjnego CC1352R1 firmy Texas Instruments. Zadanie okazało się dosyć łatwe do zrealizowania.

Układ scalony BME680 jest jak kilka układów w jednym: czujnik ciśnienia i temperatury (jak w układzie BMP280) z rozszerzeniem o czujnik wilgotności (jak w układzie BME280) i z dodatkowym czujnikiem gazu [S34]. Czujnik wilgotności względnej jest wykonany w technologii MEMS z zastosowaniem pomiaru zmian pojemności kondensatora z polimerową folią [S32]. Czujnik ciśnienia barometrycznego jest wykonany w technologii MEMS z zastosowaniem pomiaru rezystancji cienkiej membrany. Pomiar temperatury jest wykonywany poprzez pomiar zmian napięcia diody krzemowej. Każda struktura scalona zastosowana w układzie scalonym BME680

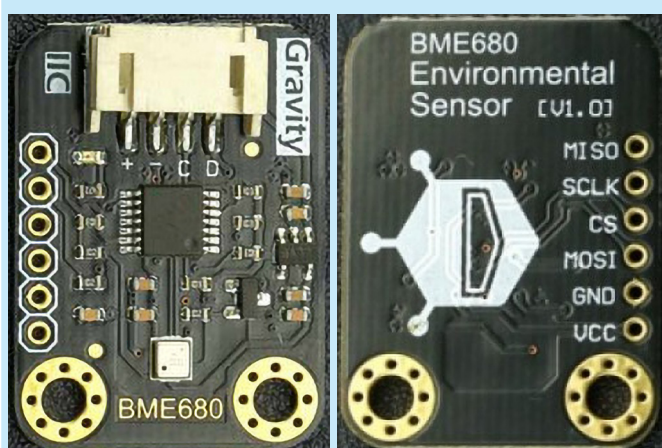
zawiera czujnik temperatury. Układ scalony BME680 został zastosowany w najnowszym zestawie Thingy:91 firmy Nordic Semiconductor przeznaczonym dla mobilnego Internetu Rzeczy [S31].

Indeks jakości powietrza IAQ

IAQ (*Indoor Air Quality*) jest liczbą która informuje o jakości powietrza we wnętrzach. Obecnie nie ma globalnego standardu definiującego IAQ [3]. Niektóre kraje mają lokalne podejście i opublikowały badania, które podają wskaźniki czystego powietrza i jego implikacje. Niestety, nie ma zgodności w zastosowaniu skali, liczby poziomów, nazw ani kolorów.

Tabela 1. Klasyfikacja jakości powietrza [5]

Wartość	Jakość powietrza	Opis
0-50	Dobra (Good)	Jakość powietrza powszechnie uznawana za satysfakcjonującą. Zanieczyszczenie powietrza na takim poziomie ma nikły lub żaden wpływ na zdrowie.
51-100	Umiarkowana (Moderate)	Jakość powietrza jest akceptowalna dla większości ludzi. Najbardziej wyczulona grupa populacji może jednak odczuwać dolegliwości.
101-150	Słaba (Little bad)	U wyczulonej części populacji mogą wystąpić negatywne skutki zdrowotne. Większa część populacji powinna czuć się relatywnie dobrze.
151-200	Niezdrowa (Unhealthy)	Cała populacja zaczyna doświadczać negatywnych skutków zdrowotnych. U najbardziej wyczulonej części populacji taki poziom może być bardzo niebezpieczny dla zdrowia.
201-300	Bardzo niezdrowa (Very Unhealthy)	Przebywanie w takich warunkach może skutkować poważnymi konsekwencjami zdrowotnymi u całości populacji.
301-500	Szkodliwa (Very bad)	Poważne zagrożenie zdrowia. Przebywanie w takich warunkach jest definitywnie szkodliwe dla zdrowia.
500+	Toksyczna (Hazardous)	W takich warunkach nie da się mieszkać. Jest to niezdrowe, niebezpieczne, nierozsądne i jeszcze kilka nie-. Zalecamy wyprowadzkę z Pekinu.



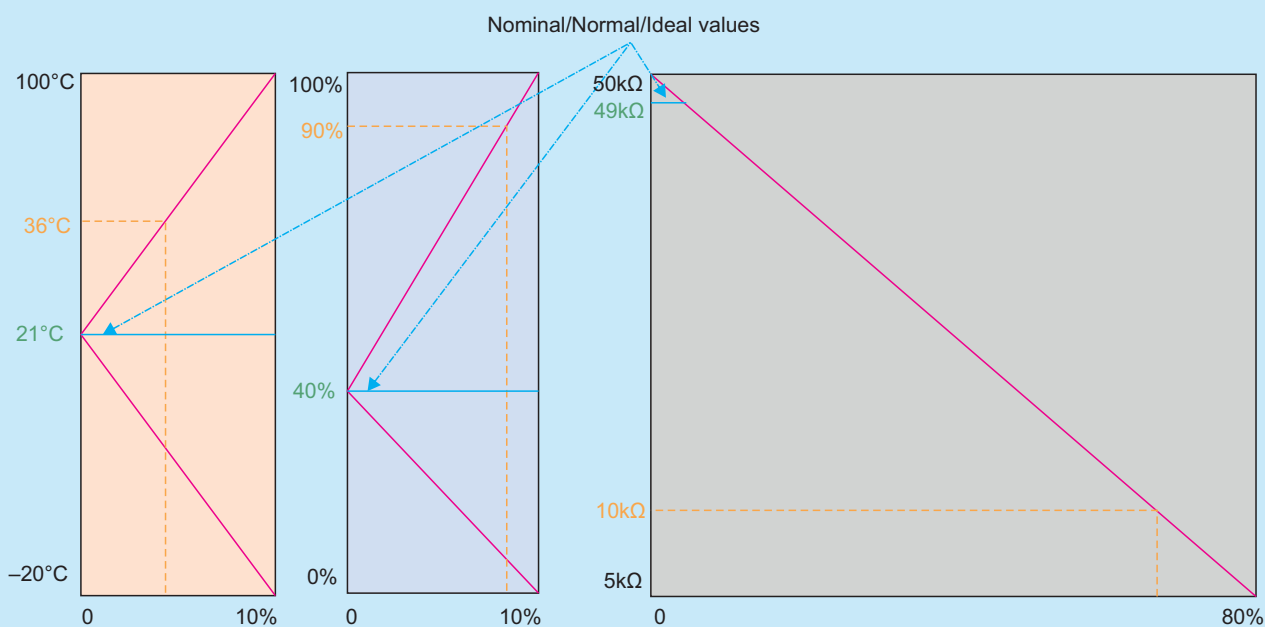
Fotografia 1. Płytkę BME680 Environmental Sensor [6]

„Umweltbundesamt” (UBA, German Federal Environmental Agency) to niemiecka agencja ochrony środowiska, która od wielu lat prowadzi badania dotyczące powietrza w pomieszczeniach i TVOC. Na podstawie badań z roku 2007 określiła standard pomiarów TVOC oraz zalecenia dotyczące jakości powietrza w pomieszczeniach

Są one bardzo przydatne, ponieważ nie tylko identyfikują poziomy potencjalnych zagrożeń, ale także dostarczają wskazówek, jakie działania są zalecane na podstawie tych poziomów TVOC [S34]. Na tej podstawie firma Bosch opracowała indeks IAQ obliczany dla czujnika BME680 przez firmową bibliotekę BSEC [2]. Klasyfikacja została pokazana w **tabeli 1**. Kolorami zaznaczony jest poziom jakości powietrza oraz dodano najczęściej stosowane nazwy angielskie.

Uproszczony sposób obliczania IAQ dla tego układu opracował i udostępnił David Bird [4]. W najprostszym wersji wskaźnik jest funkcją wilgotności, która daje 25% wartości indeksu oraz stężenia gazu, które dają pozostałe 75%. Wartości składowe są funkcją odchylenia od wartości optymalnej. Wilgotność mierzy się od 0 do 100% i jest powszechnie uznawana za optymalną, gdy wynosi 40%. Wtedy jej udział w wartości indeksu będzie wynosił 0. Wzrasta on liniowo do wartości maksymalnej przy wzroście lub obniżeniu się wilgotności.

Podobny sposób obliczania indeksu IAQ z dodatkowym uwzględnieniem poziomu temperatury został pokazany na **rysunku 1**. Stężenie gazu dla normalnego powietrza bez zanieczyszczeń (bez gazów szkodliwych) odpowiada najwyższej rezystancji wyjściowej czujnika wynoszącej 50 kΩ lub więcej. Układ BME680 zwykle odczytuje wartość rezystancji czujnika gazu w zakresie od 50 kΩ i więcej



Rysunek 1. Sposób obliczania indeksu IAQ [4]

[4]. Zakłada się, że zmiana jest liniowa i udział w wartości indeksu wynosi odpowiednio między 75% a 0%. Rezultatem sumowania udziału pomiarów wilgotności i gazu jest indeks jakościowy skalowany w zakresie 0...500. Gdzie wartość 500 oznacza złą jakość powietrza, a wartości opisowe są stosowane przedziałami od dobrej doniebezpiecznej jakości powietrza (jak w tabeli 1).

Praca układu BME680

Układ scalony BME680 pracuje w dwóch stanach: uśpienia (*sleep*) oraz pracy (*forced mode*). Tryb pracy jest ustawiany na bitach 1:0 rejestru Config (pod adresem 0x75). Wpis 00b oznacza ustawienie stanu uśpienia a wpis 01b ustawienie stanu pracy. Układ po włączeniu zasilania automatycznie ustawia tryb uśpienia. W tym stanie nie wykonywany jest pomiar i pobierana jest minimalna moc ze źródła zasilania.

W stanie pracy wykonywany jest pojedynczy pomiar w sekwencji: temperatura, ciśnienie, wilgotność, rozpoczęcie podgrzewania czujnika gazu, wykonanie pomiaru gazu i zakończenie grzania.

Można ustawić w rejestrach układu scalonego 10 konfiguracji pracy (*set-point*) czujnika gazu G0 do G9.

Na konfigurację składa się ustawienie rezystancji docelowej grzejnika, czasu podgrzewania oraz wartości prądu grzejnika. Czas podgrzewania jest liczony od momentu włączenia grzejnika do momentu rozpoczęcia pomiaru rezystencji czujnika gazu. Czas ten można ustawić w zakresie od 1 do 4032 ms. Typowo ustawia się temperaturę pracy czujnika gazu na poziom pomiędzy 200 a 400°C. Wewnętrzna pętla sterowania grzejnika pracuje z uwzględnieniem zmian rezystancji grzejnika. Program użytkownika musi na podstawie wymaganej temperatury wyliczyć ustawienia rejestrów sterujących. W dokumentacji podane są wzory obliczeniowe oraz wskazane są rejestry z wartościami parametrów do tych obliczeń [2]. Po zakończeniu pomiaru gazu układ automatycznie powraca do stanu uśpienia.

Sposób pracy i odpowiednie wzory obliczeniowe dla pomiaru temperatury, ciśnienia i wilgotności są opisane w literaturze [2]. Odczyt danych pomiarowych trzeba rozpocząć od odczytu rejestru stanu *Eas_status_0* (0x1D) z polami bitowymi:

- 7 – *New data*, dane nowego pomiaru są zapisane w rejestrach,
- 6 – *Gas measuring*, zmiana stanu na zero sygnalizuje zakończenie pomiaru gazu,
- 5 – *Measuring*, zmiana stanu na zero sygnalizuje zakończenie wszystkich pomiarów,
- 3:0 – *Gas meas index*, numer konfiguracji pracy czujnika gazu.

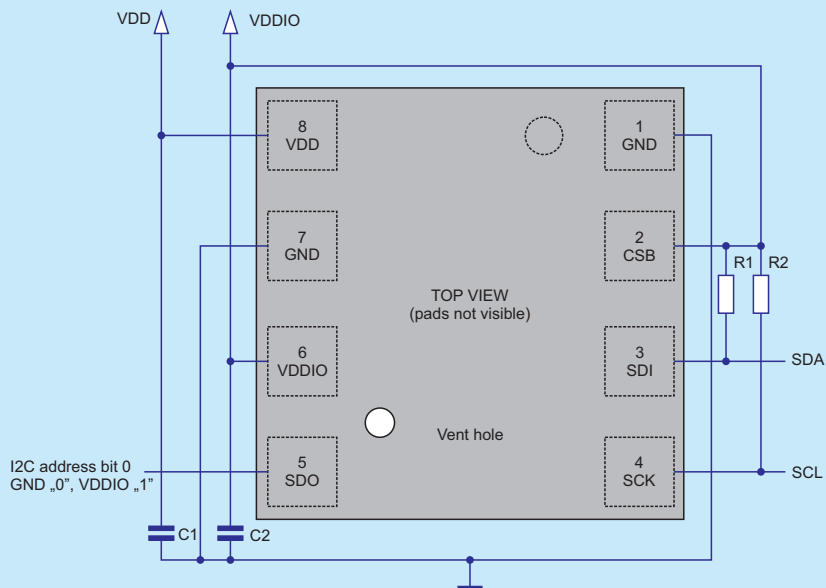
Następnie trzeba sprawdzić stan bitów *gas_valid* oraz *heat_stab* rejestru *gas_valid_r* (0x2B) aby ustalić czy dane pomiarowe są poprawne oraz czy grzejnik pracował stabilnie.

Odczytane dane z rejestrów danych układu należy przeliczyć z uwzględnieniem danych z rejestrów kalibracyjnych układu.

W przypadku wykonywania przez układ BME680 pomiaru gazu włączany jest wewnętrzny grzejnik czujnika gazu. W stabilnych warunkach długiej sekwencji pomiarów powoduje to wystąpienie stałego podwyższenia mierzonej temperatury oraz obniżonego odczytu wilgotności. Można to zjawisko skompensować poprzez uwzględnienie stałej poprawki lub odczytu z dodatkowych czujników zewnętrznych.

Rejestry wewnętrzne

Układ zawiera wiele 8-bitowych rejestrów: kalibracji, sterowania, danych, statusu oraz rejestry zarezerwowane. Dostępna jest przestrzeń adresowa z zakresu 0x00 do 0xFF. Rejestry mają inny adres przy dostępie poprzez interfejs I²C oraz SPI:



Rysunek 2. Konfiguracja wyprowadzeń podczas pracy w standardzie I²C [2]

- Podczas pracy z interfejsem I²C stosowany jest adres 8-bitowy z dostępem do całej przestrzeni adresowej,
- Podczas pracy z interfejsem SPI stosowany jest adres 7-bitowy z dostępem dwóch stron wybieranych przez ustawienia rejestru *spi_mem_page*. Po włączeniu zasilania wybrana jest strona zerowa (0x80 do 0xFF). Wpisanie do rejestru *spi_mem_page* wartości 1 powoduje wybranie strony 1 (0x00 do 0x7F).

Mapa i opis rejestrów jest zamieszczony w dokumencie BME680 Datasheet [2]. Mapa nie zawiera wszystkich rejestrów, a opis nie zawiera omówienia niektórych istotnych rejestrów układu.

Interfejs komunikacyjny

Interfejs komunikacyjny układu BME680 ma osobne wyprowadzenie zasilania VDDIO. Poziom wysoki sygnałów wejściowych interfejsu cyfrowego odnosi się do poziomu napięcia na wyprowadzeniu VDDIO [2]. Układ może pracować w trzech konfiguracjach wyprowadzeń (w standardzie: I²C, SPI (4 linie) lub SPI (3 linie)):

- Wyprowadzenie SCK pracuje zawsze jako wejście,
- Wyprowadzenie SDI pracuje jako wejście (SPI 4 linie), wejście/wyjście SPI (3 linie) lub jako linia dwukierunkowa danych (I²C),
- Wyprowadzenie SDO pracuje jako wyjście (SPI 4 linie) lub jako wejście wyboru adresu (I²C),
- Wyprowadzenie CSB pracuje zawsze jako wejście. Pełni rolę wejścia wyboru (chip select) dla pracy SPI.

Wybór standardu komunikacyjnego jest wykonywany na podstawie stanu wejścia CSB. Jeśli podczas włączenia zasilania układu wyprowadzenie CSB jest dołączone do napięcia VDDIO (podczas włączania zasilania układu) to wybierany jest standard I²C. Poziom niższy niż VDDIO spowoduje trwałą pracę ze standardem SPI. Po podaniu na wejście CSB poziomu niskiego (nawet jeden impuls) aktywowany jest na stałe interfejs SPI.

Interfejs I²C

Układ obsługuje cyfrową komunikację w standardzie I²C oraz SPI jako układ „slave”, która pozwala na odczyt lub wpis do rejestrów. Podczas pracy układu BME680 w standardzie I²C do komunikacji używane są trzy wyprowadzenia (rysunek 2):

- SCK – jako wejście zegara, linia SCL,
- SDI – jako dwukierunkowa linia danych SDA, musi być wewnętrznie dołączona do napięcia zasilania poprzez rezystor podciągający (min. 4,7 kΩ).
- SDO – jako wejście ustawiające wartość bitu LSB adresu:

rozpoczyna się od wystawienia przez układ Master (M) stanu Start (S) i 7-bitowego adresu układu (0x76 lub 0x77) z bitem RW=0 (write), czyli w trybie zapisu. Po wystawieniu przez układ BME680 (Slave, SL) bitu potwierdzenia (Acknowledge, ACKS) master wysyła bajt adresu rejestru. Po otrzymaniu potwierdzenia ACKS Master wystawia stan Strat (lub Restart) z 7-bitowym adresem układu i bitem RW=1 (read), czyli w trybie odczytu. Układ BME680 (Slave, SL) wystawia bit potwierdzenia (ACKS) oraz bajt danych.

Układ Master może:

- wystawić potwierdzenie (ACKS) – wtedy układ wyśle bajt odczytany z rejestru następnym wyższym adresem (autoinkrementacja),
- wystawić NACKM (*not acknowledge*) – wtedy zostanie zakończona transmisja.

Przykład odczytu sekwencji bajtów jest pokazany dalej na **rysunku 12**.

Biblioteka Bosch Sensortec drivers

Firmowe drivery dla układu BME680 są dostępne na stronie Bosch Sensortec drivers [10].

Repozytorium BoschSensortec Github udostępnia dwa komplety drajwerów:

- BME680 sensor driver/API – drivery z opisem i przykładami,
- BSEC-Arduino – biblioteka do Arduino dla integracji biblioteki firmowej BSEC.

Pakiet programowy BME680 sensor driver/API zapewnia niskopoziomą komunikację z układem BME680 oraz kalibrację wartości odczytu z sensorów. Uwalnia to użytkownika od konieczności zajmowania obsługa poszczególnych rejestrów układu scalonego.

Pobierany z repozytorium plik *BME680_driver-master.zip* zawiera trzy pliki źródłowe:

- *bme680_defs.h* – zawiera stałe, makra i deklaracje,
- *bme680.h* – zawiera deklaracji funkcji API,
- *bme680.c* – zawiera definicje funkcji API,
- *REDAME.md* – Zawiera opis sposobu dołączania drajwerów do projektu użytkownika.

Dodatkowo jest pobierany folder *Self test* z dwoma plikami:

- *bme680_selftest.h* – zawiera stałe, makra i deklaracje funkcji,
- *bme680_selftest.c* – zawiera definicje funkcji wykonywania testu i analizy danych.

Pakiet BME680 sensor driver/API obsługuje komunikację z układem BME680 w dwóch standardach: I²C oraz SPI (4 linie). Na stronie repozytorium zamieszczony jest przykładowy kod inicjalizacji układu (dla obu standardów komunikacji), skonfigurowania układu oraz wykonania pojedynczego pomiaru i odczytania wyników.

W nocy aplikacyjnej AN-014 [3] została opisana praca układu scalonego oraz postępowanie podczas wykonywania testowania układu: weryfikacja identyfikacji struktury scalonej, weryfikacja danych kalibracyjnych, generacji CRC, pomiar danych z czujników, weryfikacja danych pomiarowych.

Biblioteka pomiarowa BSEC

Biblioteka Bosch Sensortec Environmental Cluster (BSEC) [11] jest przeznaczona do obsługi pracy z czujnikiem BME680. Komunikację oraz kompensację danych zapewnia driver firmowy [10]. Algorytmy biblioteki zapewniają również dodatkową kompensację pomiarów czujnika gazu z uwzględnieniem wilgotności oraz dryftu. Kod z biblioteki jest zamknięty. Pobierany z repozytorium plik *bsec_1-4-7-4_generic_release.zip* zawiera pliki źródłowe oraz dokument „*Integration Guide, Bosch Software Environmental Cluster (BSEC)*” z dokładnym opisem.

Biblioteka BSEC udostępnia cztery tryby pracy z czujnikiem [średni pobór prądu]:

- Tryb bardzo niskiego poboru mocy – pomiary są wykonywane co 300 s (0,09 mA),

- Tryb szybki bardzo niskiego poboru mocy – pomiary T, P, H są wykonywane co 3 s, a pomiar gazu co 300 s (0,1 mA),
- Tryb niskiego poboru mocy – pomiary są wykonywane co 3 s (0,9 mA),
- Tryb ciągły – pomiary są wykonywane co 1 s (raczej do testowania) (12 mA).

Dla każdego trybu pracy są osobne konfiguracje dla pracy z napięciem zasilania 1,8 V oraz 3,3 V. Dodatkowo dla każdej konfiguracji możliwa jest kompensacja długoterminowa z uwzględnieniem okresu 4 lub 28 dni.

Biblioteka BSEC udostępnia kilka istotnych parametrów:

- Surowa wartość rezystancji – wartość pobrana z czujnika,
- Skompensowana wartość rezystancji [Ω] – wartość obliczona z uwzględnieniem temperatury i wilgotności,
- Indeks IAQ (IAQ) – wartości 0...500, używany algorytm automatycznego trymowania,
- Statyczne IAQ (s-IAQ) – bez trymowania,
- Poziom CO₂ (eCO₂) [ppm] – estymowany poziom CO₂ na podstawie poziomu VOC,
- Poziom VOC (b-VOC) [ppm] – estymowany poziom VOC dla powietrza wydechane (breath),
- Poziom gazu [%] – aktualny poziom rezystancji odniesiony do zapamiętanego najniższego odczytu (0%) oraz najwyższego odczytu (100%).

Moduł czujnikowy BME680 Environmental Sensor

Kilka firm oferuje moduły czujnikowe z układem scalonym BME680. Typowo zawierają one układ scalony BME680 oraz dodatkowo (opcjonalnie) układy interfejsu cyfrowego oraz regulator napięcia zasilania. Typowym przykładem jest moduł czujnikowy BME680 Environmental Sensor (DFRobot SEN0248) [6] oferowany również przez Kamami [7].

Opis płytki zestawu oraz przykładowy kod programu jest zamieszczony na firmowej stronie Wiki [8]. Schemat modułu jest dostępny ze strony Kamami [7]. Układ scalony BME680 (U3) jest zasilany z napięcia 3V3 dostarczanego przez regulator napięcia LP5907. Napięcie jest stałe i wynosi 3,3 V. Do tego napięcia dołączona jest dioda LED, sygnalizująca obecność zasilania. Układ regulatora jest zasilany z napięcia wejściowego VCC o wartości maksymalnej 5,5 V. Wartość minimalna wynosi ok. 3,3 V. Napięcie VCC zasilania modułu może być pobierane ze złącza P1 lub P2 (**rysunek 5**). Wyprowadzenia cyfrowe układu scalonego BME680 (U3) są dołączone do układów konwersji poziomów logicznych.

Sygnały komunikacji ze standardem SPI są dołączane do złącza P2 płytki. Sygnały wejściowe CSB oraz SCK układu U3 (odpowiednio SPI_CS i SPI_SCK na złączu P2) są dołączone do wyjścia bufora SN74HC125 (U2) zasilanego z napięcia 3V3. Po stronie napięcia VCC mają one dołączone rezystory podciągania 10 kΩ. Wejście SDI (odpowiednio SPI_MOSI) jest dołączone do wyjścia bufora U2 poprzez szeregowy rezystor 1 kΩ.

Wyprowadzenie SDO układu scalonego poprzez szeregowy rezystor 1 kΩ jest dołączone do złącza P2.1 (MISO). Ma ono dołączony rezystor podciągający 10 kΩ do napięcia 3,3 V. Przy pracy układu scalonego z obsługą komunikacji I²C wyprowadzenie SDO jest wejściem adresowym. Rezystor podciągający ustawia jego stan na wysoki co oznacza wybór adresu 0x77 na szynie I²C.

Sygnały komunikacji ze standardem I²C są dołączane do złącza P1 płytki. Obie linie mają rezystory podciągające 10 kΩ dołączone do napięcia VCC. Sygnał SCK_5V tego złącza jest bezpośrednio podawany na wejście bufora SN74HC125 (U2) i dalej do wejścia SCK układu U3. Sygnał dwukierunkowy SDI_5V jest ze złącza P1 podawany na układ konwersji poziomów zrealizowanego na tranzystorze Q1 i dalej do wejścia SDI układu U3.

Bardzo podobnie zorganizowany jest moduł Adafruit BME680 [9]. Jest ciekawy ze względu na rozbudowane wsparcie programowe dla modułu tej firmy. Kolejny bardzo „minimalistyczny” moduł czujnika BME680 oferuje firma Nientigo [5].

Zestaw startowy CC1352R1 LaunchPad (Texas Instruments)

Zestaw startowy SimpleLink Multi-Band CC1352R Wireless MCU LaunchPad Development Kit (LAUNCHXL-CC1352R1) jest wyposażony w układ scalony CC1352R1F3 [S15]. Zestaw jest przeznaczony do pracy w pasmach 868 MHz/915 MHz oraz 2,4 GHz. Cała elektronika zestawu CC1352R1 LaunchPad jest umieszczona na jednej wielowarstwowej płytce drukowanej. Płytkę jest podzielona na dwie części: u góry płytki jest emulator sprzętowy standardu XDS110 oraz układ monitora zasilania a na dole układ scalony CC1352R.

Na złączach J1...J2, umieszczonych po obu bokach płytki drukowanej, jest udostępnionych 26 wyprowadzeń wejścia-wyjścia (GPIO) układu scalonego CC1352R oznaczane „DIOxx”, gdzie xx-numery od 01 do 31. Dodatkowo na dole płytki są umieszczone dwa potrójne złącza z wyprowadzoną masą „GND” oraz zasilaniem „3V3”.

Połączenia sprzętowe

W komplecie z płytką BME680 Environmental Sensor jest dostarczony przewód połączeniowy Gravity, z wtyczką pasującą do złącza P1 płytki. Po drugiej stronie przewodów są zamontowane pojedyncze wtyczki żeńskie IDC. Dołączanie przewodów do zestawu startowego CC1352R LaunchPad [S15]:

- Przewód czarny (GND, P1.3) na GND (P1.3) płytki CC1352R1 LP
- Przewód czerwony (3V3, P1.4) na 3,3V (J1.1) płytki CC1352R1 LP
- Przewód zielony (SDI, P1.1) na DIO05 (J1.19, SDA) płytki CC1352R1 LP
- Przewód niebieski (SCK, P1.2) na DIO04 (J1.18, SCL) płytki CC1352R1 LP

Sposób podłączenia kabelków jest pokazany na runku tytułowym. Dodatkowo jest tam pokazany zestaw uruchomieniowy SHT31 Smart Gadget firmy Sensirion zastosowany do referencyjnego pomiaru temperatury i wilgotności [S32].

Pakiet programowy SimpleLink Sensor and Actuator Plugin (Texas Instruments)

Do obsługi układu BME280 z zastosowaniem różnych procesorów służy darmowy pakiet programowy SimpleLink Sensor and Actuator Plugin firmy Texas Instruments [13]. Pakiet zawiera przykładowy projekt *i2cbme280*, który korzysta z firmowego pakietu BME280 sensor driver/

API firmy Bosch. Jest on kompatybilny z pakietem BME680 sensor driver/API. Trochę zamieszania wprowadza stosowanie w opisach pakietu zamiennie nowej nazwy SimpleLink Sensor and Actuator Plugin (nowej) oraz starej nazwy Sensor Actuator Interface Layer (SAIL). Na przykład tak nazywa się folder z zainstalowanym pakietem na dysku.

Pakiet SimpleLink Sensor and Actuator Plugin (1.50.00.00) ma następujące wymagania [13]:

- Windows 7, Windows 8, Windows 10, Ubuntu 14.04, Mac OS X El Capitan,
- Code Composer Studio 9.2, lepiej zastosować najnowszą wersję CCS v9.3,
- `simplelink_cc13x2_26x2_sdk_3_40_00_02` SDK, lub nowsza wersja [12].

Tworzenie projektu obsługi czujnika BME680

Poniżej zostanie opisany sposób dołączenia czujnika BME680 do procesora CC1352R1. Czujnik jest zamontowany na płytce drukowanej modułu BME680 Environmental Sensor [6], a procesor na płytce drukowanej zestawu startowego CC1352R1 LaunchPad [S15]. Jako środowisko programowe zostanie zastosowany Code Composer Studio w najnowszej wersji (9.3). Można go darmowo pobrać ze strony producenta Texas Instruments. Po zainstalowaniu CCS trzeba doinstalować pakiet SimpleLink Sensor and Actuator Plugin (1.50.00.00) [13] oraz pakiet SimpleLink cc13x2 26x2 SDK (3.40.00.02) [12].

Pracę rozpoczynamy od użycia projektu *i2cbme280_CC1352R1_LAUNCHXL_tirtos_ccs* z pakietu *SimpleLink Sensor and Actuator Plugin*. Projekt jest przeznaczony do obsługi czujnika BME280. Ma on taki sam interfejs programowy jak czujnik BME680. Różni się trochę zawartością mapy adresowej rejestrów oraz dodaniem pomiaru gazu. Program wysyła napisy poprzez łącze UART do portu wirtualnego COM komputera dołączonego do zestawu startowego CC1352R1 LaunchPad. Umożliwia oglądanie na bieżąco aktualnych danych pobranych z czujnika. Opis projektu i parametrów jest zamieszczony w pliku *REDAME.html* w folderze projektu.

1. Dołącz zestaw startowy CC1352R1 LaunchPad kablem USB do komputera.
2. Wystartuj środowisko CCS.
3. Utwórz folder roboczy o nazwie *work_BME680* w ścieżce `C:\home_dir`. Taki sposób nazywania i umieszczania folderu roboczego ułatwia potem orientację wśród wielu folderów o podobnych nazwach. Umożliwia też przenoszenie foldera roboczego projektu pomiędzy komputerami o tej samej strukturze folderów.
4. Zaimportuj projekt *i2cbme280_CC1352R1_LAUNCHXL_tirtos_ccs* z pakietu SimpleLink Sensor and Actuator Plugin.

Wybrane artykuły kursu „Systemy dla Internetu Rzeczy” w miesięczniku Elektronika Praktyczna

- [S15] Zestaw CC1352R1 LaunchPad, EP5/2018
 [S21] Zestaw Nordic Thingy:52 IoT Sensor Kit, EP12/2018
 [S31] Nordic Thingy:91 – platforma prototypowania dla mobilnego IoT, EP11/2019

Literatura

- [1] BME680 Low power gas, pressure, temperature & humidity sensor, Bosch Sensortec, <http://bit.ly/2P0cyry>
 [2] BME680 Datasheet, rev.1.3, July 2019, BST-BME680-DS001, Bosch, <http://bit.ly/2U9wiL1>
 [3] BME680 Self-test application note, rev.1.4, May 2018, BST-BME-680-AN014, Bosch, <http://bit.ly/2WfzQJK>
 [4] BME680 IAQ Example, David Bird, <http://bit.ly/2TXVvcD>
 [5] Moduł czujnika BME680 – v.1.1a – Shield do Wemos D1 mini, Nientigo, <http://bit.ly/2UdmTSD>
 [6] Gravity: I2C BME680 Environmental Sensor, SKU:SEN0248, Product Page, DFRobot, <http://bit.ly/2x72Pta>
 [7] Gravity: I2C BME680 Environmental Sensor – czujnik środowiska 4 w 1, DFRobot SEN0248, Kamami, <http://bit.ly/2wdWvAb>

- [S32] Scalone czujniki cyfrowe wilgotności i temperatury, EP 12/2019
 [S34] Cyfrowe czujniki gazu, EP 2/2020

- [8] DFRobot's BME680 Environmental Sensor, Wiki, <http://bit.ly/2w5Ly3P>
 [9] Adafruit BME680, Adafruit, <http://bit.ly/2QpkD9M>
 [10] BoschSensortec, Github, BME680 sensor driver/API including example guide. Bosch Sensortec GmbH, 29 Jun 2018, <http://bit.ly/3d6plDf>
 [11] Bosch Sensortec Environmental Cluster (BSEC) v1.4.7.4, July 3rd, 2019, <http://bit.ly/2Qod2by>
 [12] SimpleLink CC13x2 and CC26x2 software development kit (SDK) (Ver. 3.40.00.02), Texas Instruments, <http://bit.ly/3818Txc>
 [13] SimpleLink Sensor and Actuator Plugin (1.50.00.00), Texas Instruments, <http://bit.ly/2w11hm2>

```

318 /* Trigger the measurement */
319 rslt = bme680_set_sensor_mode(&gas_sensor);
320 if (rslt == BME680_OK)
321     Display_print0(display, 0, 0, "BME680 Trigger the measurement OK!");
322 else
323     Display_print0(display, 0, 0, "BME680 Trigger the measurement Error");
324
325 /* Get the total measurement duration so as to sleep or wait till the
326 * measurement is complete */
327 uint16_t meas_period; /* Wake up duration in ms */
328 bme680_get_profile_dur(&meas_period, &gas_sensor);
329
330 Display_print2(display, 0, 0, "Heater temperature: %d Celsius degree, Heater duration: %d ms", gas_sensor.gas_sett.heatr_temp, gas_sensor.gas_sett.heatr_dur);
331 Display_print1(display, 0, 0, "Ambient temperature: %d", gas_sensor.amb_temp);
332 Display_print1(display, 0, 0, "Measurement duration: %d ms\n", meas_period);
333
334 user_delay_ms(meas_period); /* Delay till the measurement is ready */
335
336 struct bme680_field_data data; /*Sensor field data structure */
337
338 /* BME680 Reading all sensor data */
339 while(1)
340 {
341     /* This API reads the pressure, temperature and humidity and gas data
342     * from the sensor, compensates the data and store it in the bme680_data structure */
343     rslt = bme680_get_sensor_data(&data, &gas_sensor);
344     if (rslt != BME680_OK)
345         Display_print0(display, 0, 0, "BME680 get data Error");
346
347     if(data.status & BME680_GASM_VALID_MSK)
348     {
349         if (data.gas_resistance > gas_resistance_max)
350             if (first) first = 0;
351             else
352                 gas_resistance_max = data.gas_resistance;
353         Display_print5(display, 0, 0, "Pressure: %4.2f hPa, Temp: %4.2f DegC, Humidity: %4.3f %RH, Gas_res: %u Ohms, Gas_res_max: %u Ohms",
354             ((float) data.pressure)/100, ((float) data.temperature)/100, ((float) data.humidity)/1000, data.gas_resistance, gas_resistance_max);
355         getIAQ(&data);
356     }
357     else
358         /* Avoid using measurements from an unstable heating setup */
359         Display_print0(display, 0, 0, "FASM Valid mask Error");
360
361     /* Trigger the next measurement if you would like to read data out continuously */
362     if (gas_sensor.power_mode == BME680_FORCED_MODE) {
363         rslt = bme680_set_sensor_mode(&gas_sensor);
364         if (rslt != BME680_OK)
365             Display_print0(display, 0, 0, "Trigger the next measurement Error");
366     }
367     sleep(1);
368 }

```

Rysunek 6. Główna pętla pomiarowa w funkcji mainThread

- Zmień nazwę projektu na `i2cbme680_CC1352R1_LAUNCHXL_tirtos_ccs`. W oknie *Project Explorer* kliknij prawym klawiszem na linię projektu, z podręcznego menu wybierz *Rename* i zmień nazwę.
- Pobierz pliki drajwera firmowego *Bosch Sensortec drivers* z repozytorium Github [10]. Na stronie repozytorium kliknij przycisk *Clone or download* a następnie *Download ZIP*. Rozpakuj archiwum w eksploratorze plików.
- Dodaj pliki do projektu. Po kolei zaznacz pliki źródłowe i przeciągnij (drag and drop) na linię projektu `i2cbme680_CC1352R1_LAUNCHXL_tirtos_ccs` w oknie *Project Explorer*. Pozostaw wybraną opcję *Copy files*.
- Zmień nazwę pliku `i2cbme280.c` na `i2cbme680.c`. Zmodyfikuj kod źródłowy pliku `i2cbme680.c` według opisu w pliku *REDAME* pakietu drajwera firmowego *Bosch Sensortec drivers*.

- Należy zastąpić linię kodu `#include <ti/sail/bme280/bme280.h>` na `#include "bme680.h"`.
- Pod definicjami zmiennych globalnych należy umieścić kod funkcji służących do obsługi komunikacji szyną I²C: `user_delay_ms`, `user_i2c_rea` oraz `user_i2c_writ`. Są to funkcje realizowane przez użytkownika specjalnie dla zastosowanego procesora.
- Poniżej należy umieścić funkcję `getIAQ` obliczającą IAQ.
- Należy zmodyfikować funkcję główną aplikacji.

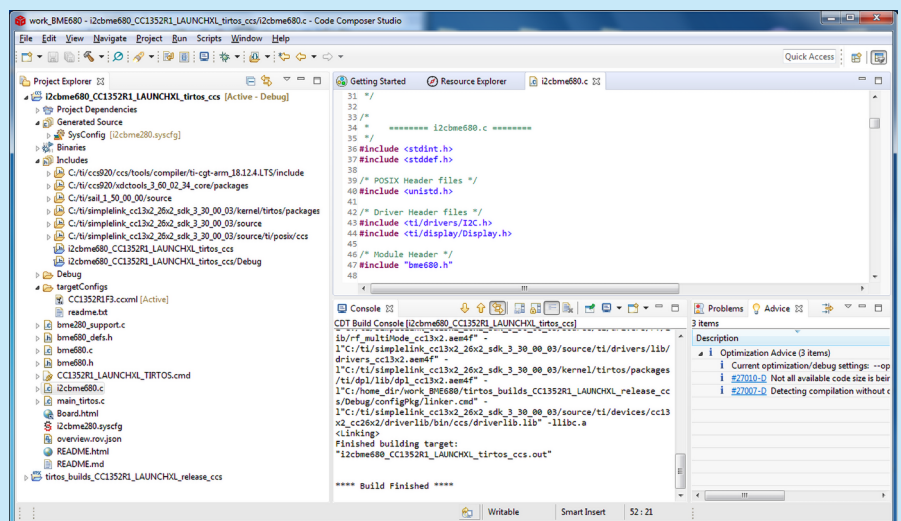
Tych modyfikacji jest sporo. Dlatego najprościej jest wymienić plik `i2cbme680.c` na pobrany ze strony FTP.

Funkcja `mainThread()` została poważnie zmodyfikowana w stosunku do wersji

Plik `i2cbme680.c` ze zmianami jest dostępny do pobrania ze strony FTP.

w pliku *readme*. Zostało dodane sporo wydruków opisujących błędy występujące podczas pracy. Bardzo ułatwia to uruchamianie programu oraz umożliwia lepsze orientowanie się w działaniu czujnika.

Pierwszy fragment kodu funkcji wykonuje inicjalizację modułu peryferyjnego I²C procesora CC1352R1. Następnie wykonywana jest inicjalizacja czujnika BME680. Dane potrzebne do konfigurowania czujnika BME680 są przechowywane w strukturze `gas_sensor` typu `bme680_dev` (zdefiniowanej w pliku `bme680_defs.h`). Po ustawieniu w strukturze parametrów czujnika wykonywana



Rysunek 7. Projekt obsługi czujnika BME680 po zbudowaniu

Listing 1. Przykładowa treść terminala podczas pracy projektu obsługi czujnika BME680

```
Starting the i2cbme680 sensor example...

I2C Initialized!
Chip ID: 0x61, Chip Addr: 0x77
BME680 Initialization OK!
BME680 sensor configuration OK!
BME680 Trigger the measurement OK!
Heater temperature: 320 Celcius degree, Heater duration: 150 ms
Ambient temperature: 20
Measurement duration: 183 ms

Pressure: 993.2299 hPa, Temp: 21.0100 DegC, Humidity: 54.0559 %RH, Gas_res: 703889 Ohms, Gas_res_max: 703889 Ohms
Hum_IAQ: 29.2836, Gas_IAQ: 124.7996, IAQ: 154
Air quality is: Unhealthy for Sensitive Groups
Pressure: 993.2500 hPa, Temp: 21.0499 DegC, Humidity: 54.1049 %RH, Gas_res: 732167 Ohms, Gas_res_max: 732167 Ohms
Hum_IAQ: 29.3857, Gas_IAQ: 114.6032, IAQ: 143
Air quality is: Moderate
...
```

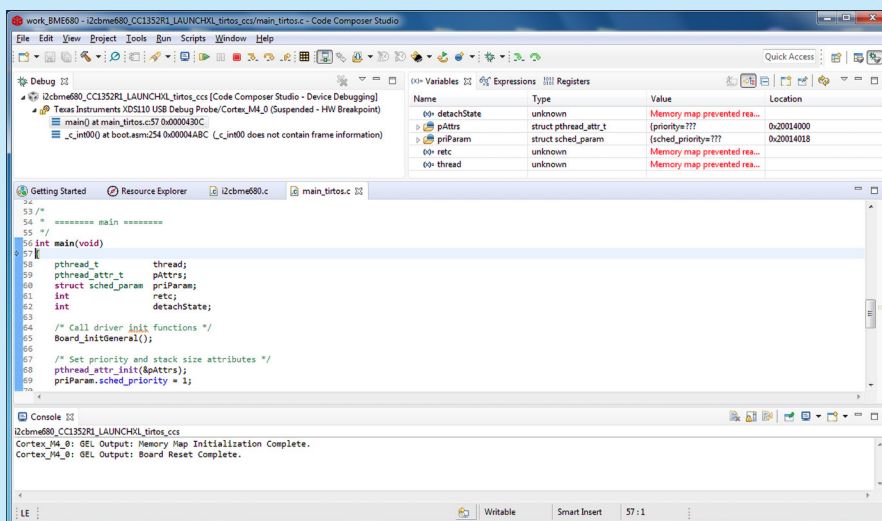
jest funkcja `bme680_init`, która wysłała do czujnika polecenie resetu programowego oraz odczytuje identyfikator układu scalonego (**rysunek 9**) a następnie dane kalibracyjne (zaczynając od adresu `0x89`, brak opisu rejestrów) oraz inne współczynniki. Następnie skonfigurowane są rejestry sterujące czujnika BME680. Ustawiane są parametry pomiaru wilgotności, ciśnienia, temperatury i gazu (w tej kolejności). Dla czujnika gazu ustawiana jest temperatura pracy na 320°C oraz czas podgrzewania na 150 ms (typowe warunki pracy).

Teraz wysyłana jest komenda (**rysunek 6**, linia 319) wystartowania pomiaru (**rysunek 10**). Następnie w funkcji `bme680_get_profile_dur` (linia 328) jest obliczany i ustawiany dokładny czas podgrzewania (profil). Dla powyższych ustawień wynosi on typowo 183 ms. Jego wartość jest bardzo ważna bo co najmniej tyle czasu trzeba po wystartowaniu pomiaru czekać na dostęp do zwalidowanych i stabilnych danych.

Praca układu jest zorganizowana w pętli nieskończonej (linia 339). Teraz już można odczytać (linia 343) skalibrowane dane pomiarowe (**rysunek 11**). Są one wysyłane do portu UART (linia 353). Następnie wysyłana jest komenda wykonania kolejnego pomiaru (linia 363) i procesor jest wprowadzany w uśpienie na czas 1 sekundy (linia 367). Po wybudzeniu procesora program powraca do wykonywania pętli nieskończonej i wysyła do portu UART dane ostatniego pomiaru. W ten sposób unika się czekania 183 ms na dostęp do danych oraz oszczędza energię zasilania.

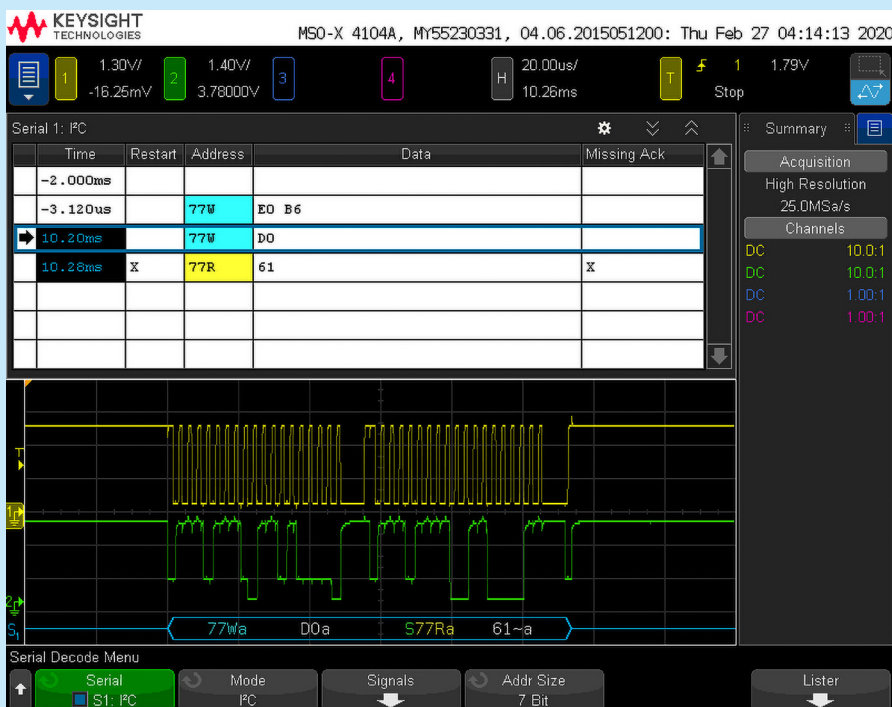
Uruchamianie projektu obsługi czujnika BME680

- Zbuduj projekt przyciskiem *Build*. Nie powinny być zasygnalizowane żadne błędy kompilacji i powinien być wygenerowany plik `i2cbme680_CC1352R1_LAUNCHXL_tirtos_ccs.out` (**rysunek 7**). Projekt zawiera już gotową definicję modułu sprzętowego (w folderze `targetConfigs`).
- Zaprogramuj przyciskiem *Debug* pamięć Flash procesora wygenerowanym kodem. Projekt jest zorganizowany z zastosowaniem systemu



Rysunek 8. Projekt obsługi czujnika BME680 po zaprogramowaniu pamięci Flash procesora CC1352R1

operacyjnego TIRTOS. Po załadowaniu kodu wykonywany jest reset procesora i program jest uruchamiany do pierwszej linii kodu funkcji `main()` w pliku `main_tirtos.c` (**rysunek 8**, linia 57). W funkcji `main` wykonywana jest inicjalizacja procesora (linia 65) i tworzone jest zadanie (wątek) `mainThread` (z kodem



Rysunek 9. Odczyt rejestru ID układu BME680

w pliku *i2cbme680.c*). Na koniec działania funkcji `main()` rozpoczyna działać system operacyjny i zostaje uruchamiane wykonanie jedyne zadania (`task`) `mainThread` (**rysunek 6**).

11. Sprawdź w *Menedżerze Urządzeń* systemu Windows jaki numer portu COM został przypisany do kanału XDS110 Class Application/User UART. Do tego portu dołącz terminal znakowy, na przykład PuTTY. Wymagane ustawienia 115200/8/1/N.
12. Rozpocznij wykonywanie programu przyciskiem *Resume*.

Przykładowy wydruk podczas pracy projektu obsługi czujnika BME680 pokazuje **listing 1**. Wartości temperatury, ciśnienia i wilgotności są pokazywane (nadmiarowo) z czterema miejscami po przecinku ponieważ wystąpiły kłopoty z obsługą formatowania wydruku liczb ułamkowych przez bibliotekę firmową.

Uruchamianie obsługi czujnika BME680

Projekt jest bardzo łatwy do debugowania w środowisku CCS. Po uruchomieniu działania programu można go zatrzymać w każdej chwili przyciskiem *Suspend*. Teraz powrót do stanu bezpośrednio po załadowaniu kodu programu (przyciskiem *Debug*) jest prosty. Trzeba kliknąć na ikonkę *System Reset* a następnie ikonkę *Restat*. Po wykonaniu restartu działanie programu jest ponownie zatrzymywane w pierwszej linii kodu funkcji `main()`.

W każdej linii kodu (nie komentarza) w plikach *ic2bme60.c* oraz *bme680.c* można ustawiać pułapki (sprzętowe). Wystarczy dwukrotnie kliknąć na pasku z numeracją linii kodu źródłowego. Dostępna jest też praca krokowa z wchodzeniem do wywoływanych w linii kodu funkcji lub bez. Najechanie kursorem na nazwę funkcji z przyciśniętym klawiszem CTRL pozwala, po kliknięciu na nią, na przeniesienie się do miejsca definicji tej funkcji. Tak samo jest z deklaracjami zmiennych i stałych.

Uruchamianie pracy czujnika BME680 okazało się dosyć trudne. Bardzo w tym pomógł oscyloskop z analizą protokołu I²C. Odczyt rejestru identyfikatora układu scalonego 0x61 spod adresu 0xD0 jest pokazany na **rysunku 9**. Najpierw na szynie I²C jest pod adres układu 0x77 wysyłany w trybie zapisu adres rejestru ID (0xD0). Następnie w trybie odczytu jest pobierana zawartość (0x61). Potwierdzenia odbioru (a – acknowledge) wystawia układ scalony (slave). Kończy transmisję *Not acknowledge* (~a) wystawiane przez procesor (master).

Wystartowanie pomiaru jest bardziej złożone (**rysunek 10**). Najpierw odczytywany jest stan rejestru `Ctrl_meas` (adr. 0x74). Pole bitowe `mode1:0` tego rejestru musi mieć wartość 0x00, co oznacza, że nie jest wykonywany pomiar. Dopiero teraz można w pole mode wpisać wartość 0x01, bez zmiany stanu pozostałych bitów.

Sekwencja odczytu danych pomiarowych trwa 517 μs, a czas powtarzania ok. 1008 ms (**rysunek 11**). Najpierw na szynie I²C jest pod adres układu 0x77 wysyłany w trybie zapisu adres rejestru `Eas_status_0` (0x1D). Następnie w trybie odczytu jest pobierana zawartość (0x80). Oznacza to, że są gotowe nowe dane. W kolejnym rejestrze (0x1E) są następne znaczniki, ale już nie ma ich opisu. O ich znaczeniu można wnioskować z kodu biblioteki firmowej. Zawartość kolejnych rejestrów jest już lepiej opisana.

Eksperymenty z obsługą czujnika BME680

Zrealizowany zestaw był eksploatowany przez dłuższy czas. Do referencyjnego

pomiaru temperatury i wilgotności został zastosowany zestaw uruchomieniowy SHT31 Smart Gadget firmy Sensirion zastosowany [S32]. Wartości odczytane z czujnika BME680 były porównywane z pomiarami wykonywanymi przez czujnik SHT31 firmy Sensirion. Typowo temperatura odczytywana przez czujnik jest wyższa od temperatury otoczenia o ok. 1,2°C, a odczytywana wilgotność niższa o ok. 2,3%.

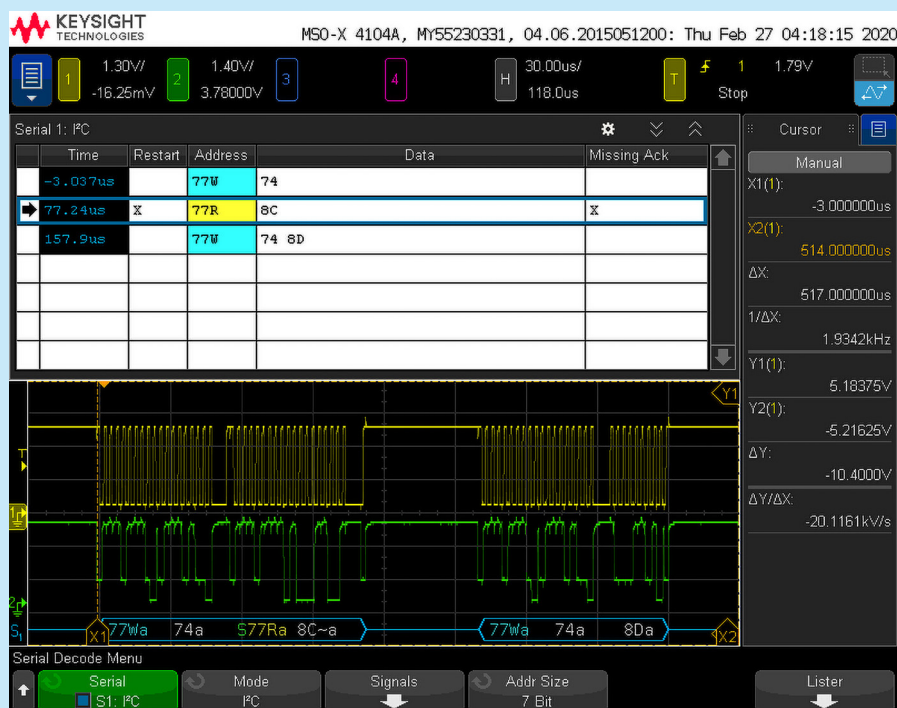
Drugi zastosowany do porównania moduł czujnikowy to Thingy:52 firmy Nordic Semiconductor [S21]. Zastosowano w nim czujnik ciśnienia LPS22HB firmy ST Microelectronics, czujnik wilgotności względnej i temperatury HTS221 firmy ST Microelectronics oraz jakości powietrza CCS811 firmy AMS. Układ CCS811 pozwala na wykrywanie w powietrzu bardzo niskiego stężenia dwutlenku węgla (CO₂) oraz lotnych związków organicznych VOC. Na podstawie rezystancji czujnika MOX obliczany jest pomiar przez wewnętrzny mikroprocesor:

- eCO₂ – równoważny (*equivalent*) poziom dwutlenku węgla, w zakresie od 400 do 32768 ppm,
- eTVOC – równoważny całkowity (*equivalent Total*) poziom lotnych związków organicznych w zakresie 0 ppb do 32768 ppb.

Odczyt ciśnienia obu czujników były prawie identyczne. Czujnik SHT31 pokazywał najniższą temperaturę. Ma on największą dokładność i jest zamontowany na „wysepce” płytki drukowanej. Thingy:52 pokazywał najwyższą wilgotność, o ok. 3% wyższą niż czujnik SHT31. Dotyczyło to także innych egzemplarzy tego modułu czujnikowego.

Układ CCS811 pokazywał bardzo niski poziom eTVOC. Kierunek zmian tego poziomu był skorelowany ze zmianami rezystancji czujnika BME680. Jednak w trakcie prób wystąpił problem z określeniem górnego i dolnego zakresu pomiaru rezystancji czujnika gazu układu BME680. Nie udało się znaleźć takiej informacji w dokumentacji firmowej. W sieci podawana jest wartość górna ok. 50 kΩ. Badania pokazały wartości sięgające 950 kΩ, a nawet ponad 1 MΩ. Ustalenie górnej wartości jest istotne dla określenia skali do obliczeń indeksu IAQ.

Po włączeniu zasilania i rozpoczęciu wykonywania pomiarów w regularnych odstępach (tutaj 1 s) odczytywana wartość rezystancji czujnika gazu zaczyna rosnąć i stabilizuje się po pewnym



Rysunek 10. Odczyt i zapis rejestru `Ctrl_meas` (komenda wystartowania pomiaru)

czasie. W opisach podawana jest nawet konieczność odczekania 30 minut [9]. Wydaje się, że dopiero zastosowanie firmowej biblioteki BSEC i obliczanie estymowanego poziomu VOC może pozwolić na porównanie działania obu czujników jakości powietrza. Jednorazowa próba w niekontrolowanych warunkach nie może być podstawą do wyciągania wniosków.

Podsumowanie

Pierwsza próba podłączenia układu BME680 do procesora CC1352R1 została podjęta w ramach projektów studenckich przedmiotu *Systemy dla Internetu Rzeczy* na wydziale Elektroniki i Technik Informatycznych Politechniki Warszawskiej. Pozwoliła ona na opracowanie sposobu użycia firmowych driverów dla modułu I²C procesora.

Integracja tego kodu z biblioteką *Bosch Sensortec drivers* wymagała ciągłego zaglądania do kodu źródłowego. Spowodowane jest to brakiem opisu biblioteki. Na szczęście jest on dosyć dobrze skomentowany.

Dodatkową trudność sprawia brak w dokumentacji firmowej pełnego opisu wszystkich rejestrów układu BME680. O działaniu niektórych rejestrów sterujących można się dowiedzieć tylko na podstawie zachowania programu. Próby z zastosowaniem kodu funkcji *selftest* dołączonej do biblioteki *Bosch Sensortec drivers* nie bardzo się powiodły. Zostaną one ponowione przy okazji uruchamiania obsługi czujnika BME680 z zastosowaniem protokołu komunikacyjnego SPI. Nie została, na razie, podjęta próba wykorzystania firmowej biblioteki BSEC.

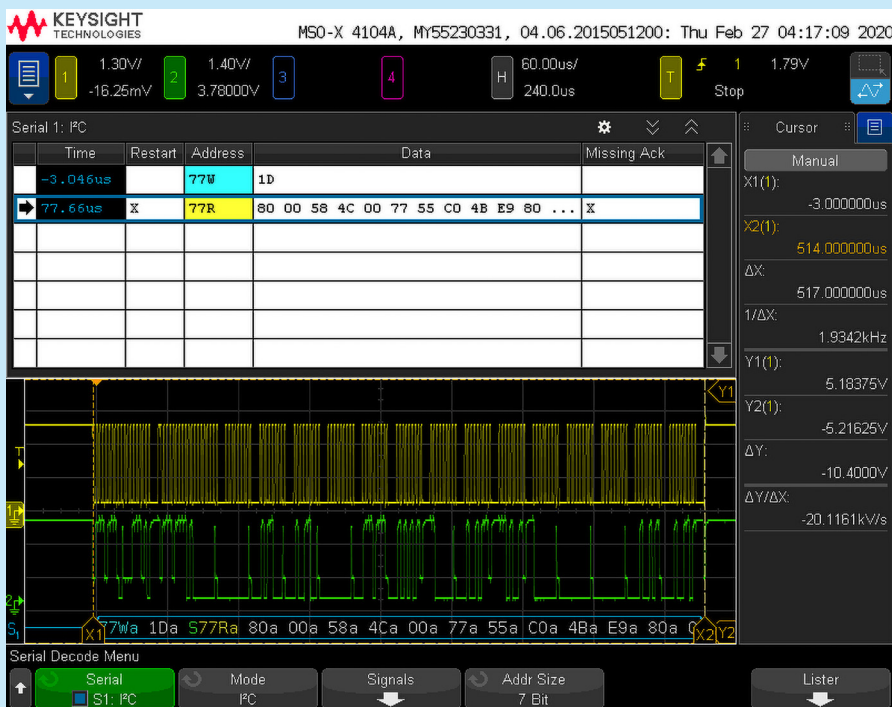
Tak jak się można było spodziewać, problemy z zastosowaniem czujnika gazu dotyczyły kalibracji. Procedury postępowania kalibracyjnego takich czujników są dosyć skomplikowane. Typowo wymagają one wystawienia czujnika na czyste powietrze. Wtedy taki minimalny poziom jest traktowany jako poziom referencyjny. Potrzebne jest do tego odpowiednie oprogramowanie i możliwość

przechowywania/zapisu danych kalibracyjnych. A i tak procedurę należy okresowo powtarzać.

Zaskakująco dobrze wygląda debugowanie działania programu. Kod biblioteki *Bosch Sensortec drivers* jest tak przejrzysty zorganizowany, że bardzo łatwo można śledzić działanie programu, nawet na poziomie pracy krokowej.

Dłuższy okres ciągłego działania układu BME680 dołączonego do procesora pokazuje stabilną pracę z występowaniem małych fluktuacji surowej wartości pomiaru rezystancji czujnika gazu. Ale aktualnym celem było tylko uzyskanie stabilnej pracy oprogramowania z użyciem procesora CC1352R1. Pozytywny wynik daje podstawę do wykonania prób dołączenia kolejnych czujników i rozbudowania oprogramowania.

Henryk A. Kowalski
Instytut Informatyki
Politechnika Warszawska



Rysunek 11. Sekwencja odczytu danych pomiarowych

Miesięcznik „Elektronika Praktyczna” (12 numerów w roku) jest wydawany przez AVT-Korporacja Sp. z o.o. we współpracy z wieloma redakcjami zagranicznymi.

Wydawca:



AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczyńska 11
tel.: 22 257 84 99, faks: 22 257 84 00

Adres redakcji:

03-197 Warszawa, ul. Leszczyńska 11
tel.: 22 257 84 60
faks: 22 257 84 00
e-mail: redakcja@ep.com.pl
www.ep.com.pl

Redaktor Naczelny:

Wiesław Marciniak

**Redaktor Programowy,
Przewodniczący Rady Programowej:**
Piotr Zbysiński

**Zastępca Redaktora Naczelnego,
Redaktor Prowadzący:**
Damian Sosnowski

**Zastępca Redaktora Naczelnego,
Menedżer Magazynu**
Marcin Karbowiczek

Szef Pracowni Konstrukcyjnej:
Grzegorz Becker, tel.: 22 257 84 58

PARK Pracownia Badań Rynku Konstruktorów:
Maksymilian Hoser, tel.: 22 257 84 65

Redaktor strony internetowej www.ep.com.pl
Dariusz Welik

Zespół marketingu i reklamy:

Katarzyna Gugąta, tel.: 22 257 84 64
Adam Kęska, tel.: 22 257 84 63
Bożena Krzykawska, tel.: 22 257 84 42
Grzegorz Krzykawski, tel.: 22 257 84 60
Maksymilian Hoser, tel.: 22 257 84 65

Sekretarz Redakcji:

Grzegorz Krzykawski, tel.: 22 257 84 60

DTP i okładka:

MAD Sp. z o.o.

Stali Współpracownicy:

Jacek Bogusz, Lucjan Bryndza, Jarostaw Doliński,
Andrzej Gawryluk, Krzysztof Górski, Tomasz Jabłoński,
Michał Kurzela, Szymon Panecki, Sławomir Skrzyński,
Ryszard Szymaniak, Adam Tatuś, Robert Wołgajew

Uwaga!

Kontakt z wymienionymi osobami jest możliwy via e-mail, według schematu: imię.nazwisko@ep.com.pl

Prenumerata w Wydawnictwie AVT
www.avt.pl/prenumerata
lub tel.: 22 257 84 22
e-mail: prenumerata@avt.pl
www.sklep.avt.pl, tel.: 22 257 84 66



Prenumerata w RUCH S.A.

www.prenumerata.ruch.com.pl

lub tel.: 801 800 803, 22 717 59 59
e-mail: prenumerata@ruch.com.pl



Wydawnictwo
AVT-Korporacja Sp. z o.o.
należy do Izby Wydawców Prasy

Copyright AVT-Korporacja Sp. z o.o.
03-197 Warszawa, ul. Leszczyńska 11

Projekty publikowane w „Elektronice Praktycznej” mogą być wykorzystywane wyłącznie do własnych potrzeb. Korzystanie z tych projektów do innych celów, zwłaszcza do działalności zarobkowej, wymaga zgody redakcji „Elektroniki Praktycznej”. Przedruk oraz umieszczenie na stronach internetowych całości lub fragmentów publikacji zamieszczonych w „Elektronice Praktycznej” jest dozwolone wyłącznie po uzyskaniu zgody redakcji. Redakcja nie odpowiada za treść reklam i ogłoszeń zamieszczonych w „Elektronice Praktycznej”.

