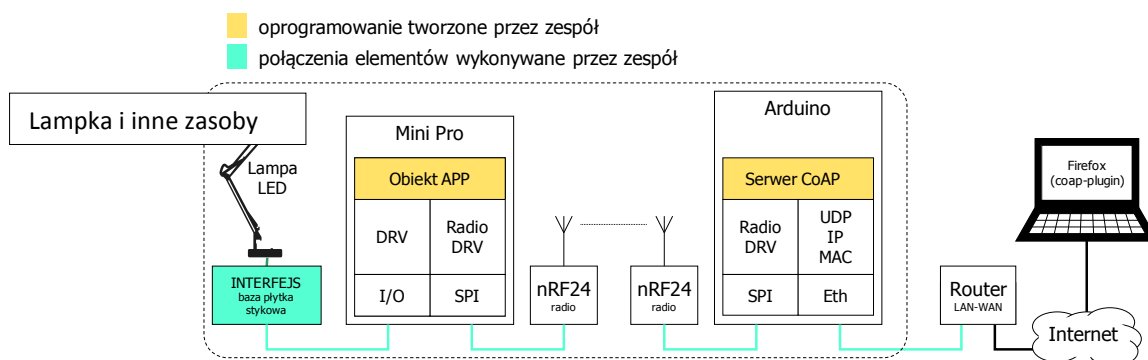


Obiekty Internetu Rzeczy (OBIR)
2019 Zima
Specyfikacja zadania projektowego

Specyfikacja ogólna:

Zrealizować system (serwer) udostępniający opisane niżej zasoby za pomocą protokołu CoAP. Należy przyjąć architekturę jak na rys. 1. W szczególności, wszystkie rozszerzenia sprzętowe powinny być przyłączone do węzła Arduino Mini Pro. Dla wspieranych zasobów należy zaprojektować URI, oraz – tam, gdzie nie jest to oczywiste lub doprecyzowane w niniejszej specyfikacji – ich stan i reprezentację. Serwer ma współpracować z podanym klientem/klientami CoAP (patrz niżej), w zakresie wynikającym z podanych niżej możliwości stworzonej przez Zespół implementacji protokołu CoAP. Serwer powinien umieć generować różne kody odpowiedzi, stosownie do sytuacji. W przypadku błędu, serwer powinien zwracać payload diagnostyczny. Do systemu należy dołączyć opisujący go dokument.

Można korzystać ze (a) „standardowych” (dostarczanych z Arduino IDE) bibliotek dla platformy Arduino, oraz (b) bibliotek, których użyto do wykonania zadań na ćwiczeniach laboratoryjnych. Można także korzystać z fragmentów kodu, użytych na ćwiczeniach laboratoryjnych (np. fragmentów zawartych w instrukcjach do ćwiczeń). Można także wykorzystać istniejącą implementację CoAPa (całą lub we fragmentach), ale w tym przypadku konieczne pełne zrozumienie pobranego z tej implementacji kodu źródłowego (ważne!). Jeśli wykorzystujemy istniejącą bibliotekę, możemy usunąć z niej elementy niepotrzebne do realizacji zadania (ze względu na ograniczoność zasobów). Poza wymienionymi elementami kodu, całe oprogramowanie niezbędne do realizacji projektu musi być stworzone samodzielnie przez Zespół projektowy. Kod źródłowy powinien być dobrze skomentowany (średnio ok. 25% linii własnego kodu powinno być opatrzone komentarzem). Wskazówka: komentowanie cudzego kodu to doskonały sposób zrozumienia go!



Rys. 1. Architektura systemu

Zakres wsparcia protokołu CoAP:

- Obsługa wiadomości NON (GET i/lub PUT, zależnie od potrzeb dla danego zasobu). Obsługa opcji Content-Format, Uri-Path, Accept. Obsługa tokena i MID.
- Obsługa opcji Observe (dla zasobów oznaczonych poniżej jako „obserwowalne”).

Obsługa wiadomości CON (GET i/lub PUT, zależnie od potrzeb dla danego zasobu) i CoAP PING. Stosowna reakcja dla zasobów oznaczonych poniżej jako „zasób o długim czasie dostępu”.

Udostępniane zasoby:

1. Zasób opisujący pozostałe zasoby. Ścieżka /.well-known/core. Ścieżki i atrybuty pozostałych zasobów powinny być określone przez Zespół. GET: pobranie reprezentacji zasobu (w reprezentacji CoRE Link Format), zasób duży (ścieżki i atrybuty należy dobrać tak, aby długość reprezentacji zasobu wynosiła ok. 60B).
2. Lampka ze ściemniaczem. Stan lampki przyjmuje wartości od 0 do 1000, gdzie 0 oznacza wyłączenie lampki, a 1000 – maksymalną jasność. GET: pobranie bieżącego stanu lampki. PUT: ustawianie lampki w zadany stan.
3. Klawiatura. Stan klawiatury to ostatnio przyciśnięty znak. GET: pobranie bieżącego stanu klawiatury w jednej z dwu reprezentacji: zwykłego tekstu (kod ASCII znaku) lub w formacie JSON. Zasób obserwowalny. (Uwaga: konieczna eliminacja drgań styków).
4. Zestaw trzech metryk (statystyk) opisujących pracę łączą radiowego między Arduino Uno a Arduino Mini Pro. Metryki powinny być zaprojektowane przez Zespół. GET: pobranie reprezentacji metryk.

Odbiór projektu:

1. Demonstracja I: interakcja z klientem CoAP według uprzednio przygotowanego przez Zespół skryptu („scenariusza”). Wszystkie wymienione w tej specyfikacji funkcjonalności powinny zostać zademonstrowane. Wymagane funkcjonalności powinny zostać zebrane w osobnej liście; przy nich powinny znaleźć się odniesienia do odpowiednich pozycji (testów) ze skryptu demonstracyjnego. Uwaga: demonstracja odbywa się z użyciem komputera prowadzących; przed rozpoczęciem własnej demonstracji, konieczne jest właściwe dla danego zespołu ustawienie działania klienta CoAP (np. czy domyślnie wysyłać żądania NON czy CON), być może pozostawionej w „niepożądanym” stanie przez poprzedni zespół; należy o tym pamiętać.
2. Demonstracja II: interakcja z klientem CoAP w sposób określony przez prowadzących.
3. Omówienie architektury oprogramowania: zaprezentowanie przez Zespół krótkiej prezentacji (5 slajdów) + dyskusja z prowadzącymi.
4. Omówienie wybranych przez prowadzących fragmentów kodu źródłowego. Uwaga: ten element odbioru ma w szczególności na celu potwierdzenie samodzielnego wytworzenia kodu.
5. Przekazanie prowadzącym płyty CD lub udostępnienie prowadzącym pendrive’u z wynikami projektu. (Uwaga: pendrive do zwrotu po trzech dniach roboczych; nie odpowiadamy za bezpieczeństwo danych na pendrive’ie.)

Zawartość płyty CD:

1. Dokument opisujący system (architektura oprogramowania: komponenty, interfejsy, przyjęte rozwiązania programistyczne, sposób testowania systemu i wykonane testy, skrypt dla celów demonstracji wraz ze wspomnianą wyżej listą funkcjonalności, zrzuty ekranu pokazujące interakcję z wtyczką CoAP).
2. Pełny, skomentowany kod źródłowy.

Kryteria oceny (w nawiasie podana liczba punktów, w sumie 40):

1. Komentarze w kodzie źródłowym (3 punkty).
2. Zapewnienie wymaganej funkcjonalności (15 punktów).
3. Sprawna i kompletna demonstracja, wg dobrze przygotowanego skryptu (6 punktów).
4. Znajomość kodu źródłowego, własnego i ew. pobranego z wybranej implementacji CoAPa (12 punktów). Uwaga: aby zaliczyć projekt, konieczne jest uzyskanie co najmniej 6 punktów ze znajomości kodu źródłowego.
5. Jakość dokumentu (4 punkty). Uwaga: dokument jest obowiązkowy. Do oceny dokumentu nie wliczamy elementów pobranych (np. dokumentacji wybranej implementacji CoAPa).

Klient CoAP

Podstawowym klientem CoAP jest wtyczka Copper (klient CoAP) dla przeglądarki Mozilla Firefox.

Uwaga: wtyczka Copper nie jest kompatybilna z aktualną wersją przeglądarki Mozilla Firefox. W projekcie należy użyć zestawu składającego się ze (a) starszej wersji przeglądarki Firefox oraz (b) wtyczki Copper. Zestaw należy pobrać stąd:

http://cygnus.tele.pw.edu.pl/olek/rozne/mmm/FirefoxPortableESR_52_6_0_CoAP.zip .

Proszę używać powyższego oprogramowania tylko do prac nad projektem OBIR. Nie należy używać go do innych zastosowań.

Dodatkowym klientem CoAP jest aplikacja CoAP Inspector, stworzona przez p. Mateusza Wojciechowskiego w ramach pracy dyplomowej magisterskiej, obronionej we wrześniu 2019 r. CoAP Inspector ma interfejs użytkownika bardzo zbliżony do interfejsu użytkownika wtyczki Copper. O ile wszystko pójdzie dobrze, udostępnimy także klienta CoAP Inspector. Zespoły, które (a) użyją obydwu klientów do współpracy ze swoim serwerem, oraz (b) udokumentują podobieństwa i/lub różnice w zachowaniu obu klientów, otrzymają bonus w postaci 5 (pięciu) dodatkowych punktów. Udokumentowanie będzie polegało na zamieszczeniu w dokumencie widoków interfejsu (screenshot) obu klientów dla tych samych operacji i wykryciu ew. różnic (dla wszystkich testów przeprowadzonych przez zespół).