Obiekty Internetu Rzeczy (OBIR)
2020 Zima
Specyfikacja zadania projektowego Z6

Zrealizować system (serwer) udostępniający opisane niżej zasoby za pomocą protokołu CoAP. Serwer powinien działać na platformie EBSimUnoEth, używanej w ćwiczeniu lab. realizowanym zdalnie. 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 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 (kodu źródłowego) 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).

Uwaga: wyraźnie zaznaczamy, które części kodu źródłowego są pobrane (a nie zrobione przez nas). Jednoznacznie identyfikujemy źródło pobranego kodu źródłowego (url, licencja). Są to warunki konieczne tego, aby praca nie była plagiatem.
Uwaga: fragmenty pobranego kodu źródłowego usuwamy poprzez ich zamianę na komentarz, a nie fizyczne usunięcie.

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 opatrzonych komentarzem). Wskazówka: komentowanie cudzego kodu to doskonały sposób zrozumienia go!

Zakres wsparcia protokołu CoAP:

1. Obsługa wiadomości NON (GET i/lub PUT, zależnie od potrzeb dla danego zasobu). Obsługa opcji Content-Format, Uri-Path, Accept. Obsługa tokena i MID.

2. Obsługa opcji Observe dla wybranej metryki z p. 3 poniżej.
   Obsługa opcji ETag.

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 formacie CoRE Link Format).
   Uwaga: jeśli Zespół ma zaimplementować opcje Block2 i Size2 (patrz wyżej), to będziemy je

testować na tym zasobie. A zatem będzie to zasób „duży” (ścieżki i atrybuty należy dobrać tak, aby długość reprezentacji zasobu wynosiła ok. 60B).

2. Graf + centrum grafu

   PUT: dodanie nowej krawędzi do grafu (nieskierowanego). Krawędź jest zadawana jako para numerów wierzchołków. Zbiór wierzchołków to {1, 2, …, N}, gdzie N to największy z numerów wierzchołków, który wystąpił we wprowadzonych krawędziach. Np. jeśli wprowadzono dwie krawędzie (1,2) i (2,5), to zakłada się, że zbiór wierzchołków to {1, 2, …, 5}. Zakłada się, że na każdym etapie dodawania krawędzi graf zachowuje spójność. Gdy wyczerpie się pamięć przydzielona na krawędzie, serwer powinien odesłać odpowiedni kod błędu. Uwaga dla ambitnych: W przypadku wyczerpania pamięci na krawędzie można w odpowiedzi umieścić opcję Size1.
   GET: pobranie wszystkich krawędzi. Uwaga: jeśli Zespół ma zaimplementować opcję Etag (patrz wyżej), to będziemy ją testować na tym żądaniu GET.
   GET: pobranie wszystkich centrów (wierzchołków centralnych) grafu. (Uwaga: centrum grafu – wierzchołek grafu spójnego taki, że największa z odległości od centrum do innych wierzchołków grafu jest najmniejsza. Źródło: Wikipedia).

3. Trzy metryki (statystyki) opisujących wymianę wiadomości/datagramów między klientem CoAP a platformą EBSimUnoEth. Metryki powinny być zaprojektowane przez Zespół.
   GET: pobranie reprezentacji metryki1.
   GET: pobranie reprezentacji metryki2.
   GET: pobranie reprezentacji metryki3.
   Uwaga: jeśli Zespół ma zaimplementować opcję Observe (patrz wyżej), to jedna z metryk powinna być obserwowalna (to na niej będziemy testować opcję Observe).
   Uwaga: jeśli Zespół ma zaimplementować obsługę żądań CON (patrz wyżej), to jedna z metryk powinna być traktowana jako „zasób o długim czasie dostępu”. W tym przypadku należy stosownie zareagować na żądanie CON, aby uniknąć retransmisji.
   Uwaga: jeśli Zespół ma zaimplementować wysyłanie odpowiedzi CON (patrz wyżej), to odpowiedzi takie powinny być generowane dla jednej z metryk. Odpowiedzi CON będziemy testować na tym zasobie.

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: przed rozpoczęciem demonstracji, konieczne jest właściwe ustawienie działania klienta CoAP (np. czy domyślnie wysyłać żądania NON czy CON).

2. Demonstracja II: interakcja z klientem CoAP w sposób określony przez prowadzących.

3. Omówienie architektury oprogramowania: zaprezentowanie przez Zespół krótkiej prezentacji (podczas prezentacji słownej pokazujemy treści, np. diagram z architekturą, z dokumentu – patrz niżej, nie tworzymy oddzielnej prezentacji) + dyskusja z prowadzącymi.