Potrzebne elementy:
- Fotooporник 5 MΩ,
- Buzzer,
- Moduł Bolt IoT Bolt WiFi,
- przewody połączeniowe.

Alarm do szafy pancernej

z podłączeniem do Internetu, zbudowany na platformie Bolt IoT

Wiele osób ma w domu szafy pancerne – niektórzy w niewielkich kasetkach trzymają kosztowności, dokumenty czy gotówkę, innym z kolei służy do przechowywania np. broni palnej. Zależnie od klasy szafy, zapewnia ona odpowiedni stopień ochrony, jeśli jednak włamywaczowi uda się w jakiś sposób ją otworzyć, to ma pełny dostęp do zawartości i nie możemy z tym nic zrobić… czy na pewno? Możliwe jest zastosowanie systemu alarmowego w takiej szafie, który może np. poinformować nas o tym, że dzieje się coś niedobrego, wysłać SMS-a na policję czy uruchomić inne systemy zabezpieczenia w domu.

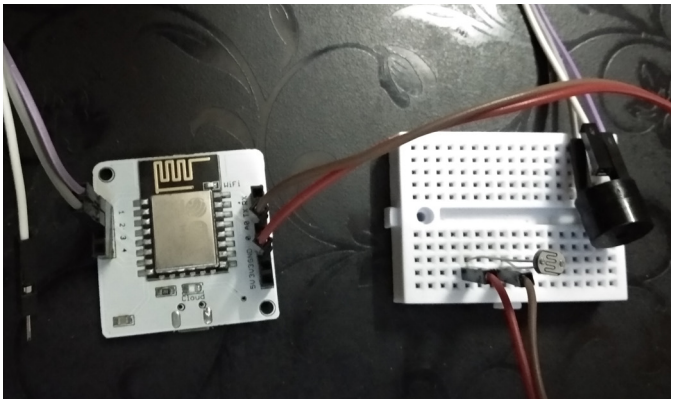
Opisywany system alarmowy przeznaczony jest do zastosowania w szafach pancernych i kasetkach praktycznie dowolnego typu. Działa na bardzo prostej zasadzie. Cały moduł umieszczony jest w sejfie, a układ wyposażony jest w zaledwie jeden sensor – fotorezystor mierzący natężenie światła we wnętrzu szafy. Gdy jest ona zamknięta, to naturalnie w jej wnętrzu jest absolutnie ciemno. W momencie gdy skrytka zostaje otwarta, do wnętrza wpada światło otoczenia. W ten sposób system wykrywa, że szafa pancerna została otwarta (zupełnie niezależnie od tego, w jaki sposób została otwarta).

Uruchomienie alarmu powoduje załączenie buzzera alarmowego podłączonego do modułu. Jeśli to byłoby za mało, system jest w stanie wysłać SMS-a na podany numer telefonu, na przykład po to, by poinformować o włamaniu właściciela czy firmę ochroniarską. Dzięki wykorzystaniu IFTTT do sterowania systemem można łatwo dodać kolejne funkcje, które będą realizowane w chwili wykrycia otwarcia drzwi.

Alarm został zbudowany na platformie Bolt IoT, która wykorzystuje m.in. narzędzia Google do analizy głosu, co pozwala na sterowanie system za pomocą prostych komend. Dzięki temu, jeśli to my otworzymy sejf, możemy natychmiastowo rozbroić alarm, wydając mu wcześniej ustaloną komendę głosową.

Podłączenie elementów elektronicznych

Buzzer i fotooporник podłączamy do modułu Bolt IoT za pomocą przewodów. Po stronie modułu sterującego muszą być one zakończone



Fotografia 1. Zmontowany układ alarmu – moduł Bolt IoT z podłączonym buzzerem i fotooopornikiem

męskimi wtyczkami w rastrze 2,54 mm (typowe goldpiny). Po drugiej stronie przewodów powinny się znaleźć złącza, pozwalające na podłączenie ww. elementów elektronicznych, zależnie od tego, jakie konkretnie zostały wybrane. Można też po prostu przylutować końcówki kabli do wyprowadzeń obu tych elementów.

Dodatni biegun buzzera podłączamy do linii zasilania, ujemny natomiast do pinu 1 modułu Bolt IoT. W ten sposób, za każdym razem, gdy na tym pinie pojawi się stan niski, różnica potencjałów na buzzerze równa będzie, w przybliżeniu, napięciu zasilania, co spowoduje, że zostanie on uruchomiony. Fotoopornik podłączony jest do pinu A0 modułu. Drugi pin fotooopornika podłączyć należy do masy. Pi A0 jest wejściem analogowym modułu, więc takie podłączenie pozwoli na pomiar napięcia na oporniku, a w konsekwencji na oszacowanie rezystancji i natężenia oświetlenia. Autor zastosował fotoopornik N5AC501085 o rezystancji nominalnej 5 MΩ i odpowiedzi spektralnej podobnej do ludzkiego oka. Można zmienić go na inny element, pod warunkiem zachowania podobnie wysokiej rezystancji nominalnej elementu i czułości na światło widzialne.

Zmontowany układ pokazano na **fotografii 1**. Czas na przygotowanie firmware i wszystkich potrzebnych komponentów programowych.

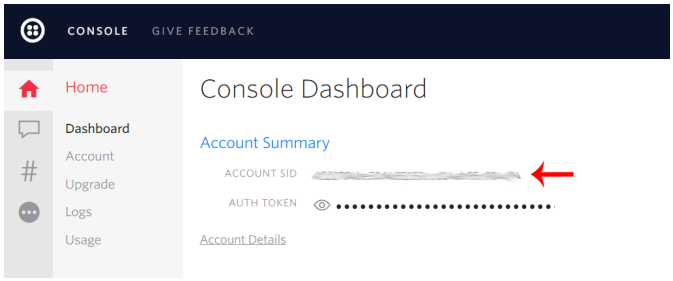
Konfiguracja oprogramowania

Oprogramowanie w opisywanym projekcie składa się z kilku istotnych elementów. Pierwszym z nich jest usługa do wysyłania SMS-ów – Twilio – którą przed wykorzystaniem należy skonfigurować. Drugim elementem jest system *If This Then That* (IFTTT), który pozwoli na połączenie asystenta Google na naszym telefonie komórkowym z systemem Bolt IoT celem przesyłania do modułu informacji o wyłączeniu alarmu. Trzecim elementem jest, pracująca w chmurze, aplikacja, która łączy wszystkie komponenty systemu w całość.

Wysyłanie SMS-ów poprzez Twilio

Aby móc wysyłać SMSy, autor wykorzystał platformę Twilio – jest to usługa wysyłania SMS-ów dostępna w chmurze. Aby z niej skorzystać, należy założyć konto na *twilio.com* oraz zakupić numer, z którego wysyłane będą wiadomości tekstowe. Wszystkie kroki opisane są poniżej:

1. Załóż konto (SIGN UP) na portalu ***http://www.twilio.com***.
2. Kup odpowiedni numer telefonu:
   a. Wejdź na stronę do zakupu numerów (*Dashboard Console* → *Phone Numbers* → *Buy a Number*).
   b. Na stronie znajduje się filtr numerów – zaznaczamy, że numer musi móc wysyłać SMS-y. Możemy dodatkowo wprowadzić część numeru, jeśli chcemy znaleźć jakiś konkretny.
   c. Klikamy *Search* (wyszukaj).
   d. Wybieramy numer, który chcemy i klikamy na *Buy* (kup).
3. Po zakupieniu numeru należy skonfigurować usługę nadawczą, wysyłającą SMS-y, gdyż ułatwi to dalszą konfigurację i zarządzanie:



Rysunek 1. Główny panel zakładki Dashbord na Twilio, gdzie znaleźć można dane do autoryzacji poprzez API

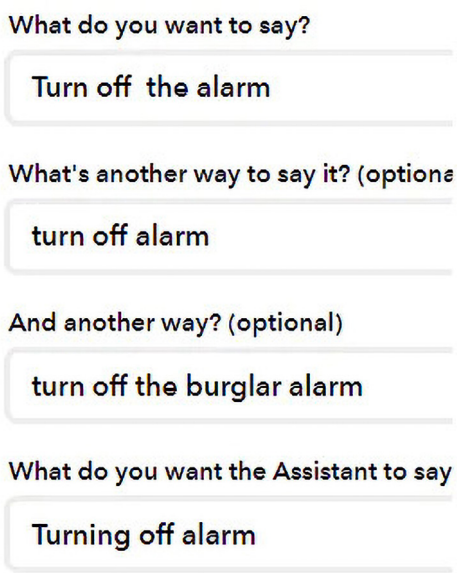
   d. Wejdź do menu zarządzania usługami (*Dashboard Console* → *Programmable SMS* → *Messaging Services*).
   e. Naciśnij *Create new Messaging Service* (utwórz nową usługę nadawczą).
   f. Nadaj nowej usłudze nazwę w polu *Friendly Name*.
   g. Naciśnij *Create* (utwórz).
   h. Zaznacz kratkę przy zakupionym wcześniej numerze.
   i. Naciśnij *Add Selected* (dodaje wybrane) – w ten sposób usługa zostanie powiązana z tym numerem.

W ten sposób skonfigurowaliśmy w pełni usługę do zautomatyzowanego wysyłania wiadomości SMS. Aby móc z niej korzystać poprzez API Twilio, musimy skopiować z naszego konta dwie informacje – SID oraz token autoryzacji, które potem umieścimy w naszej aplikacji. Informacje te znaleźć możemy na głównej stronie zakładki Dashboard, jak pokazano na **rysunku 1**.

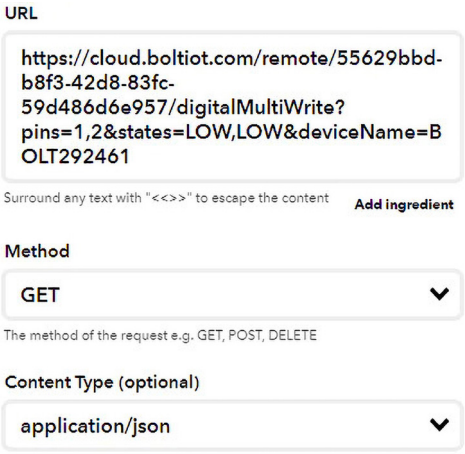
Integracja IFTTT z asystentem Google

Kolejnym elementem systemu, jaki musi zostać skonfigurowany, jest wyzwalacz w IFTTT. *If This Then That* (dosłownie „Jeżeli to, to tamto") jest usługą, która zgodnie ze swoją nazwą zapewnia możliwość tworzenia prostych wyzwalaczy – w zadanym momencie, gdy spełniony jest jakiś warunek, IFTTT, wykonuje określona w usłudze czynność. Dzięki ogromnej ilości kompatybilnych API, system IFTTT ma bardzo szerokie możliwości w komunikowaniu się pomiędzy różnymi zewnętrznymi usługami. W tym projekcie zostanie on wykorzystany do komunikacji z Google Assistant.

1. Jeśli nie posiadasz takiego konta, załóż konto na IFTTT (***https://ift.tt/2yx2dhe***). Koniecznie trzeba logować się tym samym adresem e-mail, jaki wykorzystany jest do logowania do Google Assistant.
2. Wejdź na swój profil i naciśnij *Create* (utwórz), a następie *+Trigger*, by utworzyć wyzwalacz.



Rysunek 2. Przykładowe komunikaty, wpisane w IFTTT



Rysunek 3. Przykład konfiguracji webhooka, łączącego IFTTT z chmurą IoT Bolt

3. Wybierz *Google Assistant*, by skorzystać z narzędzi do tego asystenta głosowego. Wykorzystamy opcję *Say Specific Phrase* (reaguj na konkretną frazę). W tym miejscu wpisujemy konkretne komunikaty, mające za zadanie wyłączyć alarm w szafie, które wypowiemy do naszego telefonu, gdy będziemy ją otwierać. Przykład pokazano na **rysunku 2**.
4. Naciśnij *Create Trigger* (utwórz wyzwalacz).
5. Naciśnij *+That*, aby dodać efekt, jaki ma zostać wyzwolony za pomocą komunikatu głosowego. W naszym przypadku będzie to przekazanie informacji do modułu Bolt.
6. Wybierz z menu *Webhooks* a następnie *Make a web request* (wyślij zapytanie webowe). Teraz wprowadzić trzeba URL do API naszego Bolta, ma ono format: `https://cloud.boltiot.com/remote/<API-KEY>/digitalMultiWrite?pins=1,2&states=LOW,LOW&deviceName=<DEVICE-ID>`, gdzie: API KEY to nasz klucz do API Bolta, a DEVICE ID to, oczywiście, numer ID naszego modułu.
7. Metoda dla powyższego zapytania to GET, a zawartość to *Application/json*. Przykład konfiguracji webhooka pokazano na **rysunku 3**.
8. Po wprowadzeniu wszystkich informacji naciskamy *Finish* (zakończ).

Oprogramowanie w chmurze

Ostatnim elementem systemu, jest skrypt pracujący w chmurze, który komunikuje się z modułem Bolt IoT i zarządza jego działaniem. Skrypt ten należy uruchomić na serwerze z dostępem do sieci. Najprościej jest uruchomić na serwerze wirtualnym, lub podobnej instancji. Autor skorzystał w swoim projekcie z instancji EC2 dostarczanej przez Amazon Web Services (AWS). Można oczywiście skorzystać z dowolnej innej usługi, takiej jak *digitalOceans* czy inny dowolny serwer wirtualny, szczególnie jeśli jakimś dysponujemy do pisania innych aplikacji IoT.

Listing 1. Kod źródłowy aplikacji działającej w chmurze i sterującej systemem

```
from boltiot import Bolt,Sms
import json, conf, time

mybolt = Bolt(conf.api_key,conf.device_id)
sms = Sms(conf.SID,conf.AUTH_TOKEN,conf.TO_NUMBER,conf.FROM_
NUMBER)

threshold = 10

def intruder():
    '''funkcja włączająca Buzzer'''
        mybolt.digitalWrite('1','LOW')

while True:
    '''funkcja do odczytu wartości fotooopornika'''
    response = mybolt.analogRead('A0')
    data = json.loads(response)
    current_voltage = int(data['value'])
    print(current_voltage)
    try:
        if current_voltage < threshold:
            print('current voltage dropped')
            intruder()
            print('Sending SMS via twilio')
            response = sms.send_sms('ALERT....Vault opened')
            print(str(response.status))
    except Exception as e:
            print('Error: ', e)
    time.sleep(2)
```

Listing 2. Plik nagłówkowy aplikacji z listingu 1, w którym konfigurowane są poszczególne parametry itp.

```
# Konfiguracja urządzenia
api_key = '<your_apiKey>'   # Token API aplikacji
device_id = '<your_deviceId>'   # Identyfikator urządzeia

# Konfiguracja SMów
AUTH_TOKEN = '<your_authToken>'     # Token autryzacji Twilio
SID = '<your_SID>'  # Numer SID
FROM_NUMBER = '<from_number>'       # Zwrotny numer telefonu
TO_NUMBER = '<to_number>'x      # Numer do wysłania SMS
```

Aplikacja w chmurze składa się z dwóch plików, pokazanych na **listingu 1** oraz **listingu 2**. Na pierwszym z nich znajduje się prosty skrypt, który monitoruje stan wejścia analogowego modułu Bolt. Jeśli jest ono mniejsze niż ustalony próg *threshold*, to uruchamiany jest buzzer i poprzez API Twilio wysyłany jest SMS pod zdefiniowany wcześniej numer.

Na listingu 2 zawarto wszystkie zmienne konfiguracyjne, jakie wykorzystuje skrypt. W tym miejscu wpisać należy dane do logowania do API Bolta (Klucz API i ID urządzenia), dane do logowania do Twilio (token autoryzacji oraz SID), a także numery telefonu, z którego i na który wysyłane są wiadomości SMS.

Podsumowanie

Zaprezentowany system pozwala na zabezpieczenie kasy pancernej przed niepowołanym dostępem. To proste rozwiązanie zapewni nam zdalne monitorowanie szafy pancernej czy w zasadzie dowolnego innego pojemnika. Czujnik światła zastąpić można przyciskiem bądź kontaktronem, jeśli zależy nam na zmianie zasady działania.

Nikodem Czechowski, EP

Źródło: ***https://bit.ly/3bpFClt***