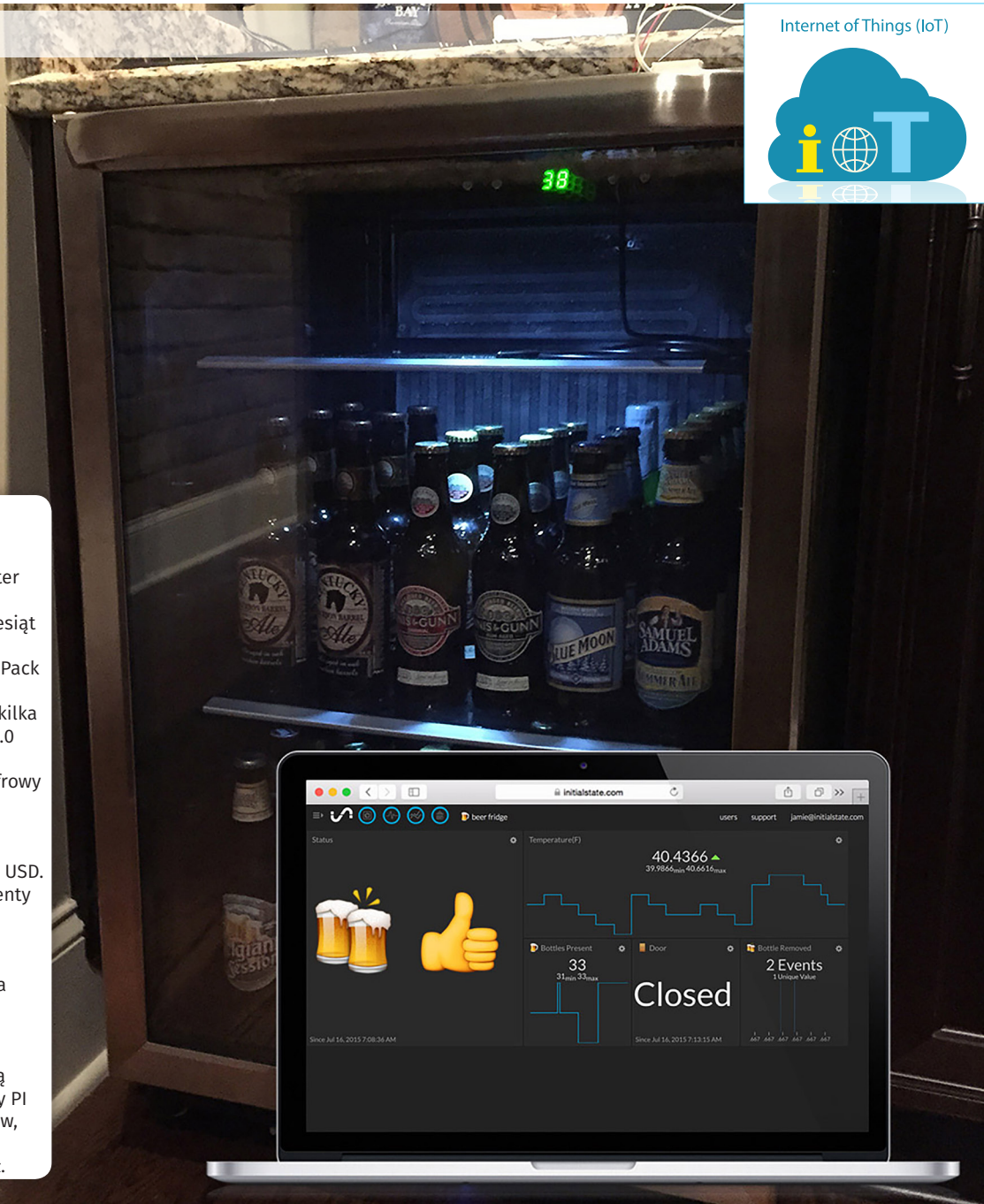




Potrzebne komponenty:

- Raspberry PI z zestawem akcesoriów (np. CanaKit Raspberry PI 2 Ultimate Starter Kit) – ok. 110 USD.
- Wii Balance Board – kilkadziesiąt USD.
- Wii Fit Rechargeable Battery Pack – około 10 USD
- Adapter Bluetooth na USB – kilka USD, w zależności czy to BT 4.0 czy starszy.
- Wodoodporny termometr cyfrowy DS18D20 – 10 USD.
- Kontaktron do montażu na drzwiach – ok. 4 USD.
- Niewielka lodówka – kilkaset USD.
- Podkładki, utówek, ew. elementy montażowe.
- Łączny koszt komponentów, po uwzględnieniu podatków, to około 3000 zł, przy czym za większość kosztu odpowiada lodówka, a jej cena może znacząco się różnić. Jeśli chce się wyposażać istniejącą lodówkę, oraz użyć Raspberry PI z minimalną liczbą akcesoriów, koszt ten powinien spaść zdecydowanie poniżej 500 zł.



Lodówka na piwo z Raspberry PI



Czy może być coś lepszego niż niewielka, specjalna, oddzielna lodówka na piwo? Takie pytanie mogłoby pewnie przejść przez myśl wielu inżynierom. My już znamy na nie odpowiedź: tak, jest to mała lodówka na piwo, wyposażona w komputer i podłączona do Internetu. Projekt takiego urządzenia, monitorowanego z użyciem Raspberry PI opracował Jamie Bailey z Nashville w USA. To dobry przykład użycia w komputera jedno płytowego w AGD oraz wykorzystania chmury do przetwarzania danych IoT.

Lodówka podłączona do Internetu była na początku XXI wieku symbolem absurdu, do jakiego można doprowadzić, gdy próbuje się skomputeryzować wszystko, co się da. Od tamtego czasu internetowe lodówki raczej

nie zdobyły zbyt dużej popularności i dopiero w ostatnich latach na rynku zaczęły się pojawiać sprzęty tego typu, których użytkowanie zaczyna mieć sens. Ciekawymi przykładami są zaprezentowane na tegorocznych targach

CES lodówki Samsunga i LG. Dzięki powstaniu internetowych sklepów spożywczych można już faktycznie rozważyć zakup lub samodzielną budowę lodówki, która będzie automatycznie zamawiała produkty przez Internet i bazując



Fotografia 1. Modem Bluetooth na USB



Fotografia 2. Wii Balance Board

na ich średnich okresach ważności, informowała o kończących się terminach przydatności pożywienia. Model Samsunga ma też wbudowane kamery, które pozwalają w trakcie zakupów w sklepie podejrzeć, jakie produkty się jeszcze nie skończyły.

Lodówka w opisywanym przypadku jest urządzeniem znacznie prostszym, a jej dodatkowe funkcje wynikają tylko i wyłącznie z ograniczenia rodzaju przechowywanych w niej produktów. Lodówka z Raspberry PI ma pozwalać przede wszystkim monitorować temperaturę, zliczać ile razy otwarto i zamknięto jej drzwiczki oraz informować o liczbie butelek znajdujących się wewnątrz w danej chwili. Co więcej, dzięki zastosowaniu serwisu do gromadzenia danych z urządzeń typu IoT, dostęp do tych informacji możliwy jest z każdego miejsca na ziemi, z telefonu, komputera czy tabletu. Dodatkowo, poprzez wbudowane w serwis internetowy funkcje analizy danych, użytkownik może przeglądać informacje historyczne i w łatwy sposób dowiedzieć się, ile piwa spożył w poszczególnych miesiącach.

Ogólna koncepcja

Urządzenie działa w następujący sposób. Wielką lodówkę należy postawić na wadze z interfejsem Bluetooth. We wnętrzu lodówki należy umieścić czujnik temperatury, a na jej obudowie kontaktron z magnesem. Wyrowadzenia z czujników należy dołączyć przewodami do komputera – w tym wypadku

Raspberry PI. Ponadto, ten komputer należy wyposażyć w interfejs Bluetooth – najwygodniej poprzez zakup modemu Bluetooth na USB (fotografia 1), dzięki czemu będzie on mógł się komunikować z wagą. Komputer należy też podłączyć do Internetu oraz skonfigurować by przysyłał dane do serwisu, w którego ramach byłyby one non-stop dostępne dla użytkownika. Ponieważ twórca projektu jest także założycielem firmy Initial State, która zajmuje się świadczeniem usług internetowych na rzecz osób z urządzeniami IoT, to właśnie serwis InitialState.com został wybrany do tego celu.

O ile pomiar temperatury oraz liczba otwarć drzwiczek zlicza się w oczywisty sposób, na podstawie wejść z czujników przewodowych, o tyle pomiar liczby butelek w lodówce nie jest już tak elegancko zrealizowany. Waga, na której umieszczana jest lodówka musi zostać wytarowana oraz należy policzyć średni ciężar jednej butelki lub puszki piwa. Następnie mierzony ciężar netto można podzielić przez średni ciężar butelki i całkiem precyzyjnie (w wypadku lodówek mieszczących do kilkudziesięciu butelek) wskazać ile piw znajduje się w danej chwili w lodówce.

Pomiar ciężaru

Największym problemem przy realizacji omawianego projektu był dobór urządzenia mierzącego ciężar lodówki. Tak jak w przypadku innych projektów, opisywanych przez nas w dziale „Projekty Soft”, samodzielna budowa takiej wagi była raczej wykluczona – autor szukał gotowego rozwiązania, które za pomocą jakiegoś uniwersalnego sposobu komunikacji mógłby podłączyć do Raspberry PI. Wybór padł na akcesorium do gier, jakim jest Nintendo Wii Balance Board (fotografia 2). Jest to kontroler do gier, mający kształt wagi łazienkowej i pozwalający na mierzenie nacisku wywieranego na powierzchnię Balance Board. W rzeczywistości kontroler pozwala również na pomiar rozkładu nacisku, ale na potrzeby wagi wystarczające będzie monitorowanie jedynie ciężaru sumarycznego. Wii Balance Board ma wbudowany interfejs Bluetooth. Koszt Wii Balance Board to około 50-60 dolarów w przypadku nowego urządzenia, ale znaleźć można też używane egzemplarze (produkt jest na rynku od 2008 roku), w cenie niemal dwukrotnie niższej.



Fotografia 4. Wii Fit Rechargeable Battery Pack – sposób na zasilanie Wii Balance Board z gniazdka elektrycznego

Wii Balance Board trudno jednak nazwać wyborem optymalnym do omawianej aplikacji. Podstawowym problemem jest konieczność każdorazowego parowania interfejsu Bluetooth wagi z Raspberry PI, po wyłączeniu zasilania. Być może nie byłoby to tak niewygodne, gdyby nie fakt, że przycisk parowania znajduje się na spodzie urządzenia, a to oznacza, że bez jakichkolwiek modyfikacji, jeśli tylko na chwilę zabraknie prądu, konieczne byłoby zdejmowanie masywnej lodówki z wagi, by odwrócić urządzenie i dokonać parowania. Dlatego autor dokonał drobnej przeróbki, w której ramach wykonał niewielką dźwignię wykonaną z ołówka, taśmy klejącej i kilku podkładek, która została tak usytuowana, by bez poruszania wagą można było nacisnąć od spodu przycisk parowania (fotografia 3).

Z użyciem Wii Balance Board wiąże się jeszcze jeden problem – jest to urządzenie zasilane bateryjnie i producent nie przewidział możliwości podłączenia wagi na stałe do prądu. Na szczęście problem ten dostrzegło wiele firm tworzących alternatywne rozwiązania dla graczy i powstały pakiety akumulatorów z wejściem do podłączenia ładowarki (fotografia 4). Pakiety te mają kształt umożliwiający podczenie ich w miejsce zatoki na baterie oraz sprawiają, że urządzenie pracuje tak, jakby korzystało bezpośrednio z prądu z sieci elektrycznej.

Ostatnim problemem z pomiarem wagi lodówki na rzeczonym urządzeniu jest fakt, że służy ono przede wszystkim do dynamicznego monitorowania nacisku, co jest ważne w grach. Tymczasem w przypadku pomiaru masy lodówki z zawartością, korzystne byłoby odfiltrowanie wszelkich drgań, powodowanych np. przez agregat chłodniczy lodówki. Autor projektu miał z tym problem, dlatego stworzony przez niego skrypt wykonuje każdorazowo po (domyślnie) 500 pomiarów, po czym podaje uśredniony wynik. Mimo to wskazania dla pustej lodówki różniły się od siebie w zakresie nawet do około 200 g.

Monitorowanie otwarcia drzwi i temperatury

Kontaktron został podłączony do wejścia GPIO17 Raspberry PI i do masy. Wejście GPIO17 zostało też podłączone przez rezystor



Fotografia 3. Autor wykonał dźwignię z ołówka, by ułatwić sobie naciskanie przycisku na spodzie Wii Balance Board

podciągający do napięcia 3,3 V (rysunek 5). Zastosowany kontaktron został tak zaprojektowany, bo można go było łatwo przymocować właśnie np. do drzwi lub okien (fotografia 6).

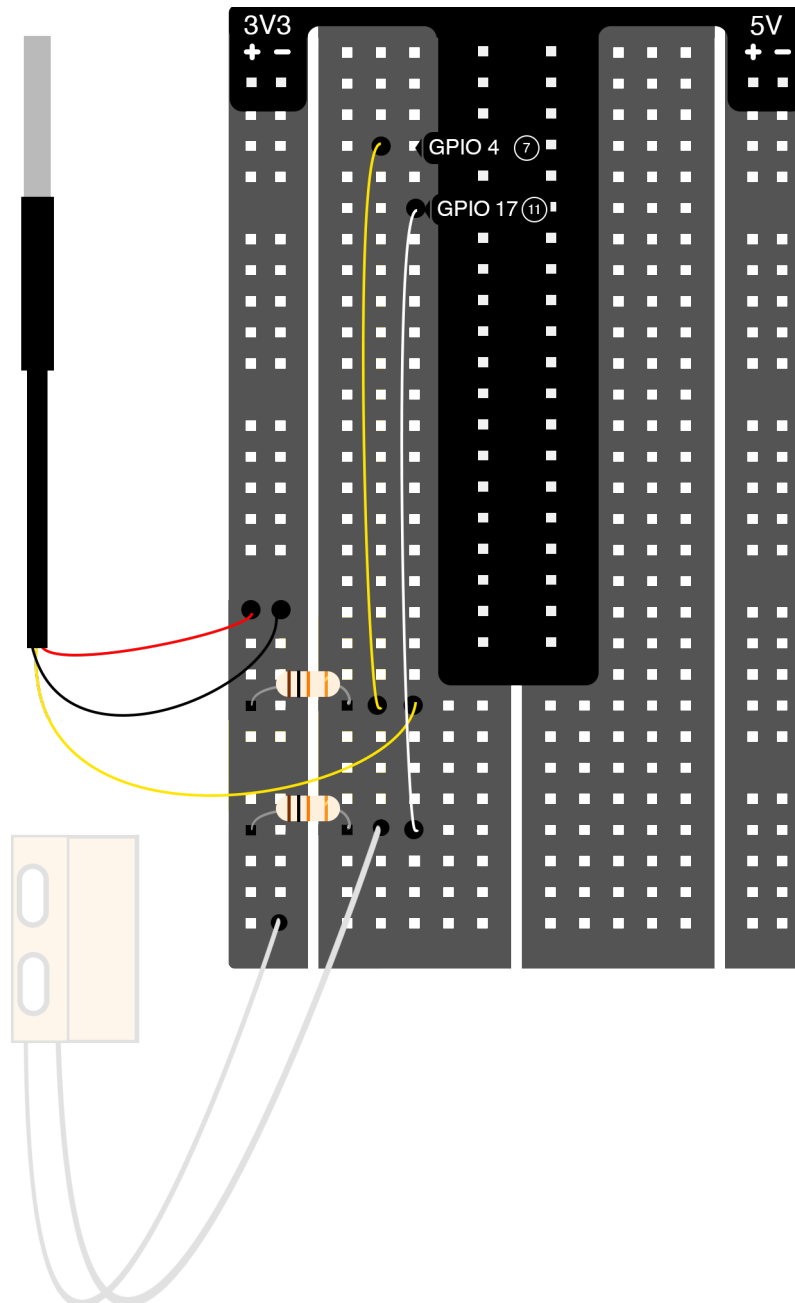
Użyty czujnik temperatury ma cyfrowy interfejs Dallas 1-wire. Został podłączony do zasilania 3,3 V oraz masy, a wyprowadzenie sygnałowe do portu GPIO4 Raspberry PI. Wejście to zostało również podciągnięte rezystorem do 3,3 V. Użyty model DS18B20 (fotografia 7) jest wodoodporny, dzięki czemu nie grozi mu skroplona w lodówce wilgoć z powietrza, a do tego może pracować w temperaturze od -55 do +125°C.

Całość została zmontowana z użyciem pomocniczej płytki drukowanej, ułatwiającej wyprowadzanie sygnałów z Raspberry PI (fotografia 8). Autor skorzystał z kompletnego zestawu CanaKit Raspberry PI 2 Ultimate Starter Kit, w którym oprócz komputera Raspberry PI 2 Model B z 1 GB pamięci RAM, znajdowała się też karta MicroSD o pojemności 8 GB, obudowa, zasilacz z wyjściem microUSB, płytki z wyprowadzeniami i taśma do nich oraz dwa 10-kiloomowe rezystory. Zestaw ten kosztuje około 110 USD + koszty transportu i podatki. Komplet użytych komponentów, bez lodówki, pokazano na fotografii 9).

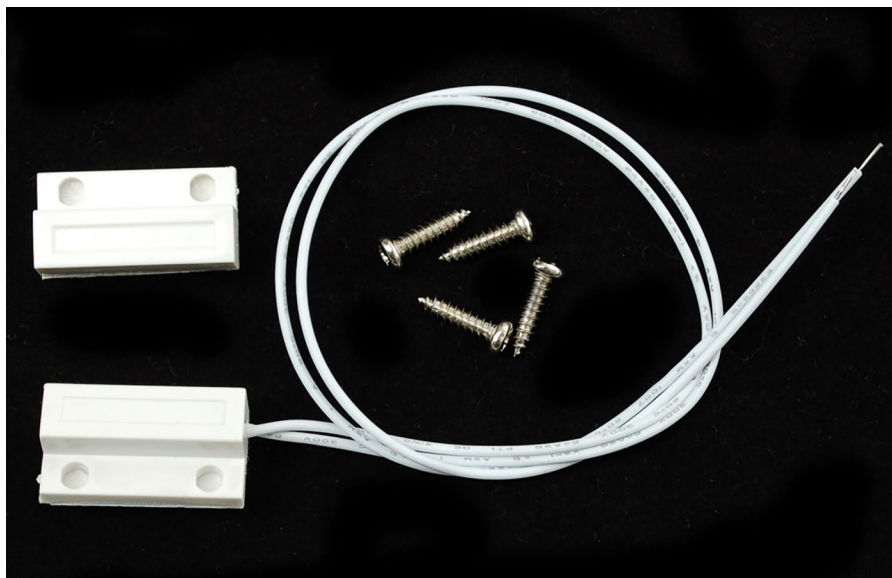
Konfiguracja

Raspbian w domyślnej postaci nie zawiera niektórych pakietów, potrzebnych w niniejszym projekcie, toteż autor musiał doinstalować odpowiednie oprogramowanie, po uprzednim zainstalowaniu Raspbiana na karcie pamięci.

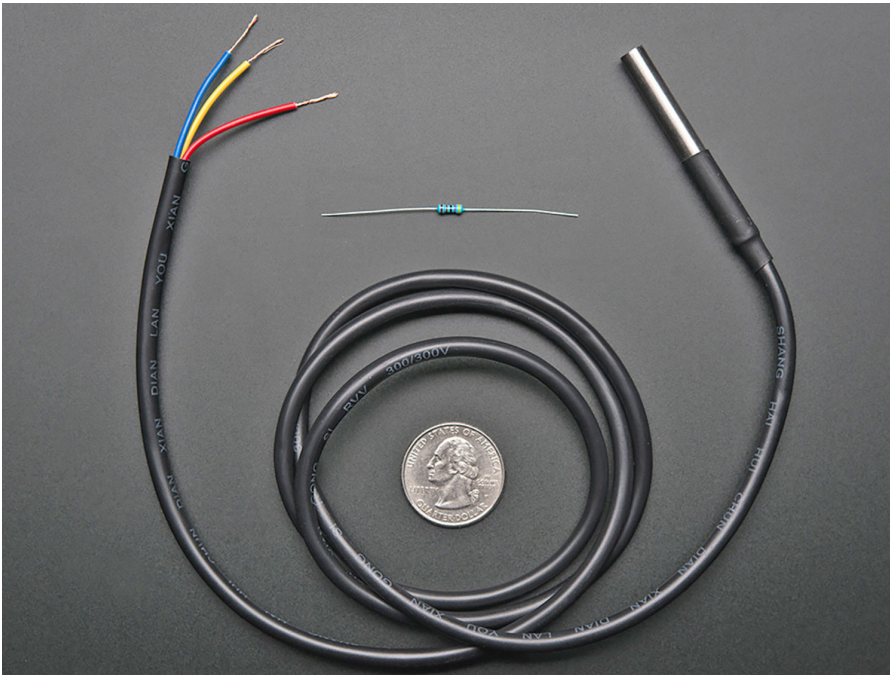
Do obsługi Bluetootha konieczne z poziomu skryptów napisanych w Pythonie konieczne było zainstalowanie pakietów **python-blueluetooth**, **bluez** i **python-gobject** (polecenie `sudo apt-get install python-blueluetooth bluez python-gobject`). Numer MAC interfejsu Bluetooth na USB można poznać korzystając z polecenia `hcitool dev`. Do sparowania wagi z Raspberry PI i korzystania z interfejsu Bluetooth, autor wykorzystał udostępniony w Internecie, zmodyfikowany skrypt, którego fragment przedstawiamy na **listingu 1**. Jest to tylko część stworzonej klasy `Wiiboard`, która służy do obsługi wagi i czytania jej wskazań. Kluczowe są procedury inicjacyjne. W głównej pętli najpierw odbywa się wyszukiwanie akcesorium i zapamiętywanie jego adresu sprzętowego. Następnie system odłącza dotychczas podłączone urządzenia Bluetooth, by te nie powodowały problemu i próbuje sparować się z urządzeniem o wcześniej pobranym adresie. W tym czasie waga musi być w trybie parowania, co oznacza że użytkownik musi wcześniej nacisnąć przycisk na spodzie wagi. Jeśli połączenie się uda, waga miga wbudowaną diodą i system przechodzi do normalnej pracy.



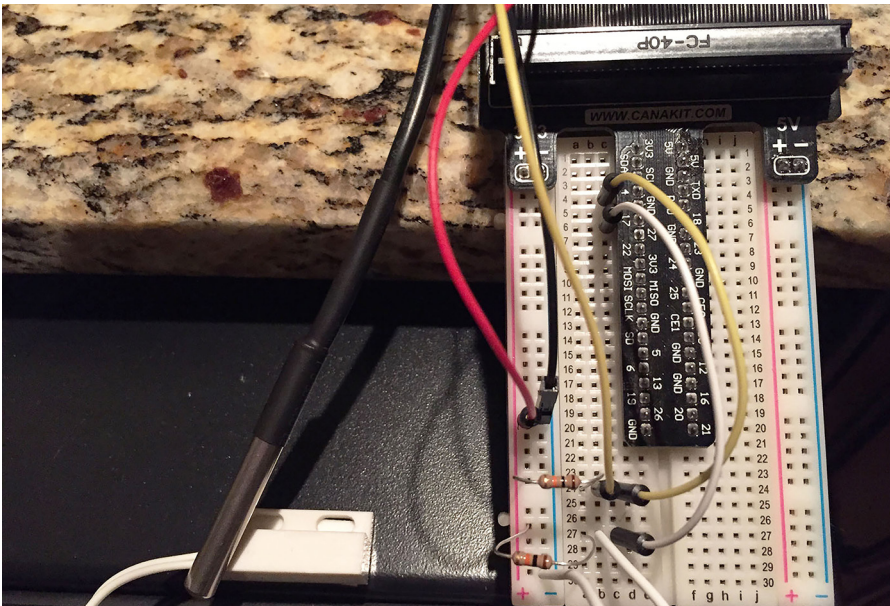
Rysunek 5. Sposób dołączenia komponentów



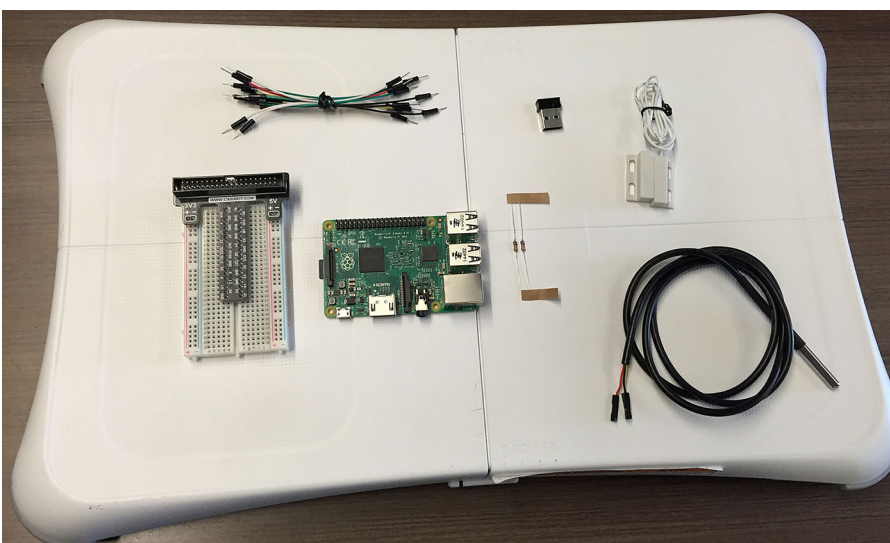
Fotografia 6. Zastosowany kontaktron



Fotografia 7. Termometr cyfrowy DS18D20



Fotografia 8. Połączone komponenty



Fotografia 9. Komponenty użyte w projekcie

Sama inicjalizacja wagi sprowadza się do zdefiniowania zmiennych i monitorowania stanu połączenia. Konieczne jest też przeprowadzenie kalibracji po nawiązaniu połączenia. Rozłączenie jest realizowane za pomocą polecenia `disconnect()`, a dane odbierane są poleceniem `receive()`.

Skrypt monitorujący otwarcie drzwi jest bardzo prosty, gdyż w pętli sprawdza stan wejścia GPIO17 i jeśli wykryje, że się on zmienił, przechodzi do obsługi tego zdarzenia. W efekcie ta część systemu nie wymaga żadnej dodatkowej konfiguracji.

Dodatkowych zmian w konfiguracji wymaga natomiast obsługa interfejsu cyfrowego używanego w czujniku temperatury. W pliku `/boot/config.txt` powinna znaleźć się linijka o treści:

```
dtoverlay=w1-gpio,gpiopin=4
```

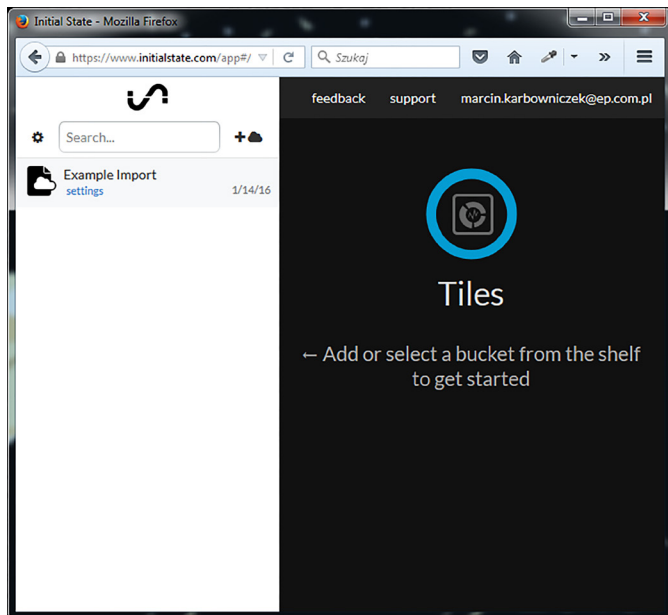
Sprawia ona, że po ponownym uruchomieniu systemu, wejście GPIO4 będzie traktowane jako interfejs 1-wire. Dzięki temu, po uruchomieniu kolejno poleceń:

```
sudo modprobe w1-gpio
sudo modprobe w1-therm
```

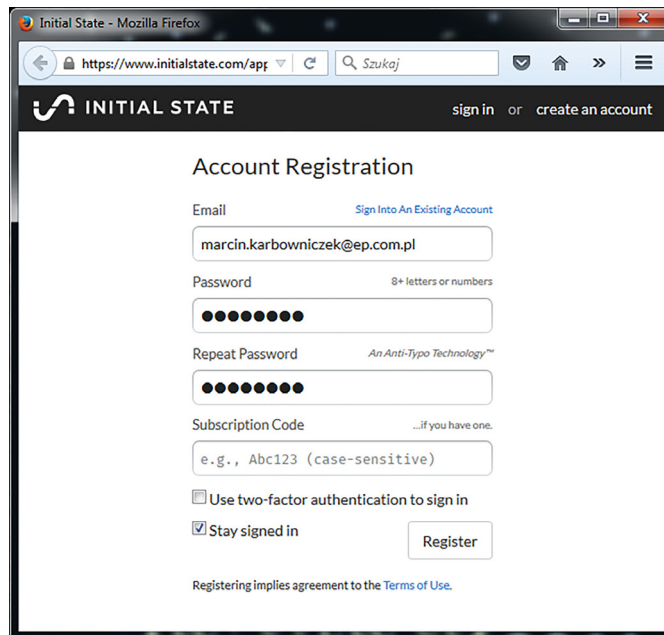
System będzie zapisywał wyniki odczytów z podłączonego czujnika temperatury do pliku. Plik ten znajduje się w katalogu `/sys/bus/w1/devices` w podkatalogu zaczynającym się od ciągu „28-”. Pełna nazwa katalogu zawiera w sobie unikalny numer seryjny sensora. Sam plik będzie nosił nazwę `w1_slave`, a w nim znajdować się będą linijki ze danymi. W naszym przypadku istotna jest tylko wartość numeryczna poprzedzona ciągiem „t=”. Jest to wskazanie temperatury, wyrażonej w tysięcznych częściach stopni Celsjusza. Oznacza to zarazem, że odczytywanie wskazania temperatury będzie wymagało ciągłego odczytywania nowych wartości ze wspomnianego pliku, co pokazano na [listingu 2](#). Oprócz konfiguracji stałych, autor stworzył w tym celu trzy funkcje. Jedna odczytuje treść pliku, druga przetwarza te dane, by pobrać temperaturę. Trzecia nadzoruje wykonywanie dwóch poprzednich w nieustannie działającej pętli, z przerwą co zadeklarowany czas.

Dołączenie do Internetu

Powyższe wskazówki właściwie pozwalają już na samodzielne przygotowanie takiej skomputeryzowanej lodówki, która korzysta z czujników i zbiera dane, ale brakuje jeszcze wygodnego sposobu prezentacji zgromadzonych informacji. W tym celu autor skorzystał z usługi <http://goo.gl/pAJSe7>, która dostępna jest w ramach trzech pakietów, w tym jednego bezpłatnego i dwóch płatnych. Pakiety płatne różnią się od siebie stopniem oferowanej pomocy technicznej, a od pakietu darmowego różnią się dopuszczalną liczbą zbieranych w ciągu miesiąca danych oraz czasem przechowywania tych danych. Głównym problemem z darmowym



Rysunek 10. Główny panel serwisu Initial State, po zalogowaniu się użytkownika



Rysunek 11. Tworzenie konta w <http://InitialState.com>

kontem jest fakt, że dane przechowywane są tylko przez jeden dzień, więc sensowne wykorzystanie takiej lodówki z użyciem darmowej subskrypcji traci sens, ale nie przeszkadza to w pokazaniu, jak korzystać z tego typu serwisów.

Serwis <http://goo.gl/pAJSe7> (rysunek 10) pozwala zrezygnować z własnego serwera WWW, uruchamianego np. na Raspberry PI i gromadzić dane od razu w chmurze. Co więcej, został przygotowany w taki sposób, by ułatwić użytkownikom korzystanie z informacji i prezentowanie ich. Dzięki temu twórca aplikacji może zaoszczędzić dużo czasu potrzebnego na samodzielnej budowę interfejsu.

Aby skorzystać z serwisu <http://goo.gl/pAJSe7> należy najpierw zarejestrować w nim konto. Proces ten trwa chwilę – wystarczy podać adres e-mail i wymyślić hasło, a opcjonalnie wskazać kod kuponu na płatną wersję usługi (rysunek 11). W celu powiązania Raspberry PI z kontem Initial State należy zainstalować skrypt ISStreamer i odpowiednio go skonfigurować. Nie będziemy wnikać we wszystkie operacje, jakie wykonuje ten skrypt, gdyż chcemy pokazać łatwość stworzenia takiej aplikacji, a nie sposób, w jaki można przygotować serwis konkurencyjny do Initial State. W ogólności wystarczy tylko wskazać, że skrypt instalacyjny musi przygotować narzędzia do komunikacji z serwerem <http://goo.gl/pAJSe7> w oparciu o login i hasło użytkownika, a ponadto musi pozwalać na przysyłanie przesyłanych danych do grup, tak by można było łatwo dzielić generowane treści z urządzeń IoT na kategorie i rozpoznawać, które pochodzą z którego komponentu systemu. W Initial State, takie grupy danych nazywa się koszami (ang. *bucket*).

Skrypt instalacyjny należy pobrać i uruchomić za pomocą komendy:

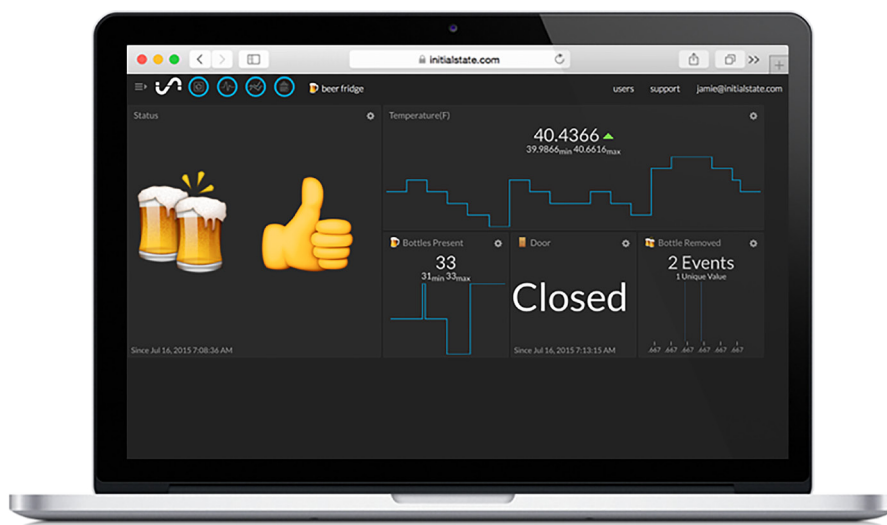
```
\curl -sSL https://get.initialstate.com/python -o | sudo bash
```

Powyższe polecenie jest równoznaczne z automatycznym wywołaniem (z uprawnieniami administratora) skryptu pobieranego z Internetu, dlatego zaleca się zarówno użycie `\curl` zamiast `curl` (aby upewnić się, że uruchomione zostanie oryginalne polecenie `curl`) oraz parametru `-sSL`, wymuszającego skorzystanie z szyfrowania podczas transmisji. Skrypt wyświetli komunikaty w stylu:

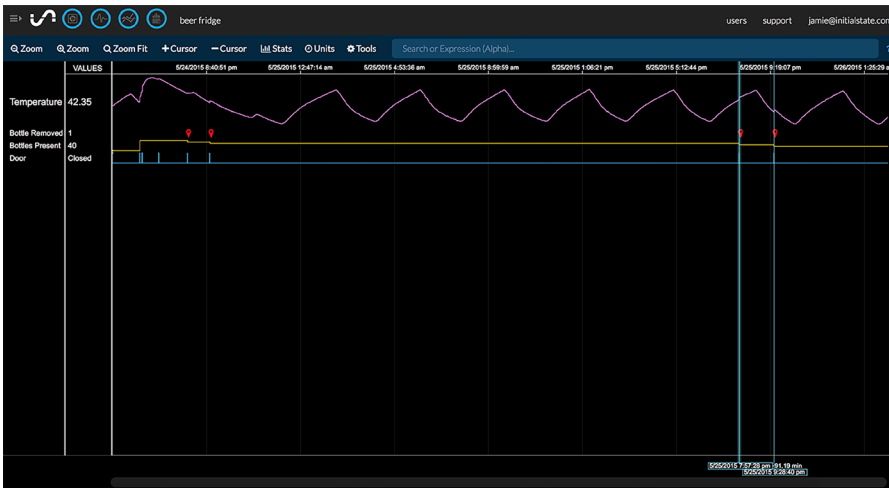
```
pi@raspberrypi ~ $ \curl -sSL https://get.initialstate.com/python -o | sudo bash
Password:
Beginning ISStreamer Python Easy Installation!
This may take a couple minutes to install, grab some coffee :)
But don't forget to come back, I'll have questions later!
```

```
Found easy_install:
setuptools 1.1.6
Found pip: pip 1.5.6 from /Library/Python/2.7/site-packages/pip-1.5.6-py2.7.egg (python 2.7)
pip major version: 1
pip minor version: 5
ISStreamer found, updating...
Requirement already up-to-date: ISStreamer in /Library/Python/2.7/site-packages
Cleaning up...
Do you want automagically get an example script? [y/N]
```

A w tym, na końcu zapyta o pobranie przykładowego skryptu wykorzystującego serwery Initial State. Można to zrobić, by sprawdzić, czy komunikacja działa poprawnie, ale nie jest to konieczne. W przykładowym pliku, a także w pełnym pliku z kodem aplikacji znajduje się polecenie z biblioteki Streamer, zainstalowanej przez wcześniej uruchomiony skrypt instalacyjny oraz



Rysunek 12. Widok typu dashboard w <http://InitialState.com>



Rysunek 13. Widok przebiegów czasowych w <http://InitialState.com>

importowanej w początkowej sekcji skryptu pythona. Biblioteka ta zawiera trzy użyteczne w tym przypadku funkcje:

Streamer(), która jest konstruktorem obiektu odpowiadającego za przesyłanie danych w chmurę,

streamer.log(), która służy do przekazywania danych do zapisu w chmurze,

streamer.flush(), która powoduje przesłanie danych na serwer Initial State i opróżnienie ich bufora.

Konstruktor przyjmuje nazwę kosza danych (`bucket_name`), klucz zabezpieczający dostęp do kosza (`bucket_key`) oraz klucz zabezpieczający dostęp do całego konta użytkownika serwisu (`access_key`). Są to stałe definiowane wcześniej w programie. Rozróżnienie pomiędzy kluczem do serwisu, a do kosza jest istotne. Klucz do serwisu nie jest hasłem użytkownika, ale niezależnie wygenerowanym ciągiem znaków. Takich ciągów znaków dla jednego konta może być wiele i mogą one jednocześnie pozwalać na dostęp do serwera. Sens używania więcej niż jednego takiego klucza związany jest z możliwością ograniczania praw dostępu. W przypadku Initial State mechanizm ten jest bardzo prymitywny – po prostu dany klucz może pozwalać na dostęp lub nie. Dzięki temu można np. za pomocą jednego konta na serwerze Initial State obsługiwać bardzo dużą liczbę urządzeń zlokalizowanych u wielu użytkowników i jeśli którykolwiek z tych użytkowników zacznie korzystać ze swojego urządzenia w nieodpowiedzialny sposób, jego klucz może zostać zablokowany, przez co straci on możliwość namieszania we wspólnym koncie chmurowym. Tymczasem kosze pozwalają na grupowanie danych – np. zbieranie informacji określonej kategorii lub pochodzących od danego użytkownika w ramach jednej grupy. I tu znowu powstaje kwestia uprawnień – osoba posiadająca klucz do konta nie może zmienić danych w koszu innej osoby, korzystającej z tego samego konta, gdyż nie posiada klucza do innego kosza niż do swojego.

Warto też wspomnieć jak działa funkcja logowania danych. W omawianym serwisie przyjmuje ona dwie zmienne, z czego pierwsza to nazwa rejestrowanego parametru, a druga to wartość.

Fragmenty kompletnego kodu znalazły się na [listingu 3](#). Jak widać, nie ma w nich żadnych poleceń odnoszących się do prezentacji danych, czy kontroli dostępu do nich. Wszystko to realizowane jest za pomocą panelu użytkownika w serwisie Initial State. Dane są przetwarzane na poziomie Raspberry PI, przygotowywane do wysyłki i przesyłane na serwer. Widać sposób, w jaki zmieniane są ikonki w panelu użytkownika (zmiana parametru „status” na wartości takie jak „:fire:”) oraz sposób posługiwania się funkcjami **streamer.log()** i **streamer.flush()**. W klasach `BoardEvent` i `Wiiboard` pokazano dodatkowe funkcje potrzebne do pobierania danych z Wii Balance Board. Kompletny kod można pobrać ze adresu <https://goo.gl/FMFGIU>.

Panel internetowy

Gotowy skrypt przesyła do serwera dane na temat aktualnej temperatury (pobierane co minutę, czyli maksymalnie 44640 wartości miesięcznie), zgłasza zdarzenia otwarcia i zamknięcia drzwi, a przy okazji – za każdym razem gdy zmieni się stan drzwi, przesyła aktualną informację o liczbie butelek wewnątrz lodówki. Panel w serwisie InitialState.com pozwala w wygodny sposób przeglądać te dane. Dostępny jest zarówno widok ogólny (tzw. dashboard – [rysunek 12](#)), jak i przebiegi czasowe ([rysunek 13](#)), pokazujące jak zmieniały się poszczególne wartości w czasie. W widoku ogólnym prezentowane są też ikonki, w symboliczny sposób pokazujące, czy aktualna temperatura w lodówce nie jest zbyt wysoka lub zbyt niska. Jest to realizowane z poziomu Raspberry PI, które za każdym razem gdy przesyła dane o temperaturze do serwera, może również zmienić treść wyświetlanego obiektu i czyni to podając inną nazwę ikonki, gdy

temperatura odbiega od zalecanej. Ewentualne dane statystyczne, np. pokazujące ile butelek zostało wypitych w ciągu miesiąca, są obliczane za pomocą mechanizmów w chmurze i tam pokazywane.

Podsumowanie

Omawiany projekt ([fotografia otwierająca artykuł](#)) trudno jednoznacznie ocenić. Z jednej strony jest to bardzo prosty mechanizm, korzystający z łatwego w użyciu, cyfrowego czujnika temperatury i jednego przełącznika oraz z cyfrowo przesyłanych informacji o wadze, a analiza gromadzonych informacji nie jest zbyt skomplikowana. Z drugiej, autor dołożył naprawdę wiele starań i wykazał się pomysłowością w odniesieniu do mechanizmu ważenia lodówki oraz parowania kontrolera do gier z Raspberry PI. Jest to jednak też bardzo pouczający przykład tego, do czego w praktyce może służyć umieszczony w chmurze serwer dla urządzeń IoT oraz jak z niego korzystać. Zgodnie z relacją autora, stworzenie projektu pozwoliło mu odkryć, że temperatura wewnątrz lodówki, dokładnie w tym rejonie, gdzie przechowywane jest piwo, regularnie waha się w zakresie 4°C, podczas gdy lodówka na swoim małym wyświetlaczu praktycznie stale pokazuje wartość średnią. Poza tym okazało się, że w trakcie nieobecności autora w domu, jeden z niepełnoletnich domowników próbował sięgnąć do lodówki, ale chyba zorientował się, że jest monitorowana i zrezygnował z wzięcia piwa.

Naszym zdaniem, projekt można rozbudować w ciekawy sposób. Zaprojektowany przez autora system umożliwia składowanie tylko jednego rodzaju napojów, a bardzo ciekawym rozwiązaniem byłoby stworzenie algorytmu, który w oparciu o te same dane jest w stanie samodzielnie rozpoznać np. dwa rodzaje trunków: choćby wina i piwa. Pomijamy fakt, że wiele z win będzie wymagało innej temperatury przechowywania niż piwo, ale można sobie wyobrazić skrypt, który zakłada że za każdym razem, gdy otwierane są drzwiczki lodówki, wyjmowana z niej lub wkładana do niej jest tylko jedna butelka (czy też puszka). Przy dokładności pomiaru zastosowanej wagi możliwe byłoby rozpoznawanie na tej podstawie, czy w lodówce ubyło/przybyło jedno piwo, czy wino, które zazwyczaj znajduje się w większych butelkach. W ten sposób komputer lodówki mógłby oddzielnie przekazywać informacje o liczbie zgromadzonych piw i win do serwera w chmurze.

Można by się też zastanowić nad zastąpieniem Wii Balance Board inną wagą – np. nowoczesną, łazienkową, podłączaną przez Bluetooth lub do sieci Wi-Fi. W praktyce jednak są to urządzenia droższe niż Balance Board.

Marcin Karbowiczek, EP