Zalanie

Czujniki zalania i wycieku, własna sonda na ESP32.

- Czujniki zalania i wycieku: rodzaje, zasada działania i zastosowania
- Czujniki zalania w praktyce: wybór, montaż i własna sonda (DIY)

Czujniki zalania i wycieku: rodzaje, zasada działania i zastosowania

Poziom: podstawy

Dotyczy: czujniki zalania i wycieku, zawory odcinające, serwerownia Zielonej, magazyn w Swarzędzu

Czas: ~40 min

Czujnik zalania mówi, że woda jest tam, gdzie nie powinna, zanim dojdzie do szafy. Na tej stronie poznajemy cztery rodzaje czujników, zawory, które potrafią zamknąć dopływ, i miejsca, w których w serwerowni woda pojawia się najpierw. Wracamy tu przy wyborze czujnika i przy planowaniu rozmieszczenia.

Skróty użyte na tej stronie (3)

Skrót	Co znaczy
ESP32	Tani mikrokontroler z Wi-Fi i Bluetooth firmy Espressif: malutki komputer, z którego budujemy własne czujniki sieciowe
SMS	Short Message Service: krótka wiadomość tekstowa na telefon, która dociera także bez internetu
SNMP	Simple Network Management Protocol: protokół odczytu stanu urządzeń sieciowych

Z życia wzięte

Z ŻYCIA WZIĘTE

Czwartek, 2:40. Telefon dyżurny budzi Tomka SMS-em z `wej01`: „Z2: wykryta woda”. W serwerowni pod klimatyzacją B stoi kałuża, bo wąż odpływu skroplin zsunął się ze złączki, a pompka tłoczy wodę prosto na podłogę. Tomek wyłącza jednostkę B wyłącznikiem serwisowym, bo jednostka A utrzyma temperaturę, i wyciera podłogę. Woda nie doszła do

> szafy R5, bo czujnik zadziałał po kilku minutach od pierwszej kałuży, a nie po dwóch godzinach, jak dawniej, gdy o wycieku dowiadywaliśmy się od sprzątaczki.

Skąd się wzięło

Pilnowanie wody jest starsze od elektryczności. Ktezybiusz z Aleksandrii, około 250 roku p.n.e., zbudował zegar wodny, w którym pływak utrzymywał stały poziom wody w zbiorniku. Historycy techniki uznają to za jeden z pierwszych samoczynnych układów regulacji, czyli urządzeń, które same reagują na zmianę poziomu [1].

Rekonstrukcja zegara wodnego Ktezybiusza z III wieku p.n.e. Pływak i zawór pilnują stałego poziomu

□□ *Rekonstrukcja zegara wodnego Ktezybiusza z III wieku p.n.e. Pływak i zawór pilnują stałego poziomu wody, a ta sama zasada działa do dziś w spłuczce i w czujniku pływakowym. Źródło: Wikimedia Commons (autor i wolna licencja podane na stronie pliku).*

Czujniki elektryczne korzystają z innego odkrycia. W 1800 roku William Nicholson i Anthony Carlisle rozłożyli wodę prądem z ogniwa Volty na wodór i tlen, a w 1834 roku Michael Faraday nazwał ten proces elektrolizą, ukuł terminy elektroda i anoda i podał dwa prawa opisujące ilość przemiany [2]. Z tego wynikają dwie rzeczy ważne dla czujnika: woda z solami przewodzi prąd, więc dwie elektrody mogą ją wykryć, a stałe napięcie na elektrodach powoduje elektrolizę i korozję.

Michael Faraday. Jego prawa elektrolizy tłumaczą, dlaczego elektrody czujnika zasilanego stale prąd

□□ *Michael Faraday. Jego prawa elektrolizy tłumaczą, dlaczego elektrody czujnika zasilanego stale prądem stałym z czasem korodują. Źródło: Wikimedia Commons (autor i wolna licencja podane na stronie pliku).*

Nie udało nam się ustalić, kto i kiedy wynalazł czujnik zalania w dzisiejszej postaci. Znamy jego składniki: przewodnictwo wody z odkryć z początku XIX wieku, pływak od Ktezybiusza i optykę całkowitego wewnętrznego odbicia. Kable sensoryczne z polimerem przewodzącym i modułem, który wskazuje miejsce wycieku, są produktami przemysłowymi, a daty ich pierwszego patentu nie potwierdziliśmy, więc jej nie podajemy [3].

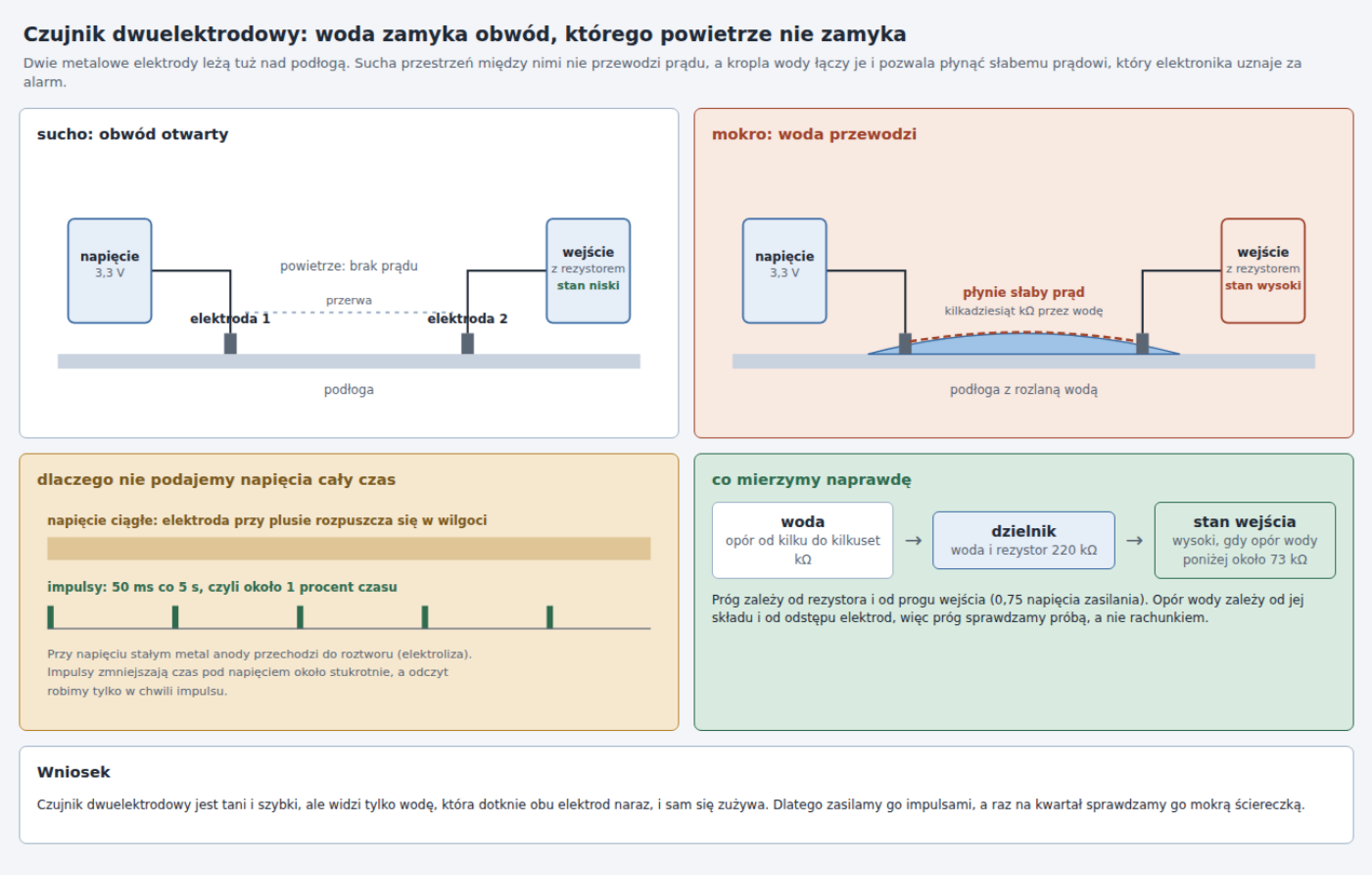
Rodzaje i zasada działania

Pojęcie: czujnik zalania. Urządzenie, które wykrywa obecność wody w miejscu, w którym jej nie oczekujemy, i zgłasza to stykiem albo sygnałem do modułu wejść. Nie mierzy ilości wody, tylko to, czy w ogóle jest.

Pojęcie: strefa i lokalizacja. Czujnik punktowy wie, że woda dotarła do jednego miejsca. Kabel sensoryczny z lokalizacją podaje dodatkowo, w którym miejscu wzdłuż kabla woda jest, a kabel bez lokalizacji mówi tylko, że gdzieś w strefie.

Punktowy dwuelektrodowy

Pojęcie: czujnik przewodnościowy. Dwie elektrody oddzielone szczeliną, między którymi w suchym stanie nie płynie prąd. Gdy kropla albo kałuża połączy je wodą, opór między elektrodami spada o kilka rzędów wielkości i elektronika zgłasza alarm.



Rysunek 1. Czujnik dwuelektrodowy na sucho i na mokro oraz różnica między zasilaniem stałym a impulsowym. Warto zauważyć, że impuls trwa ułamek procent czasu, więc elektrody prawie się nie zużywają. (kliknięcie otwiera pełny rozmiar)

Zakres i dokładność: wykrywa wodę w jednym punkcie, na elektrodach; próg czułości zależy od przewodności wody i konstrukcji elektroniki. Czas reakcji: od kilku do kilkudziesięciu sekund po zetknięciu elektrod z wodą. *Zalety: tani, prosty, łatwy do zbudowania samodzielnie, wykrywa też cienką warstwę wody.* Wady: elektrody korodują, zwłaszcza przy zasilaniu stałym, kurz i skropliny dają fałszywe alarmy, a woda o małej zawartości soli przewodzi znacznie słabiej niż woda z kranu, więc czułość trzeba dobrać do wody, której szukamy (sprawdzamy w karcie czujnika). * Gdzie: pod klimatyzatorami, w tacach skroplin, pod zmywarkami i pralkami, przy zaworach i pompkach.

Kabel sensoryczny z lokalizacją

> **Pojęcie: kabel sensoryczny.** Kabel, który ma kilka przewodów i warstwę reagującą na wodę, układany wzdłuż podłogi, ściany albo rury. Woda zmienia opór między przewodami w jednym miejscu, a moduł podłączony do kabla odczytuje zmianę jako alarm.

W wersji z lokalizacją moduł wysyła przez kabel prąd stały i mierzy, ile napięcia spada na drodze do miejsca zwarcia. Spadek jest proporcjonalny do odległości, więc moduł zamienia go na liczbę metrów od początku kabla [3]. Przykładowy kabel z polimerem przewodzącym reaguje tylko na wodę, a nie na inne ciecze, i według producenta wraca do stanu spokoju w ciągu kilkunastu sekund od wyschnięcia [3].



Rysunek 2. Kabel sensoryczny z lokalizacją: napięcie spada wzdłuż kabla, a zwarcie wodą w jednym miejscu daje napięcie, z którego moduł liczy odległość. Warto zauważyć, że moduł nie potrzebuje czujnika na każdy metr. (kliknięcie otwiera pełny rozmiar)

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

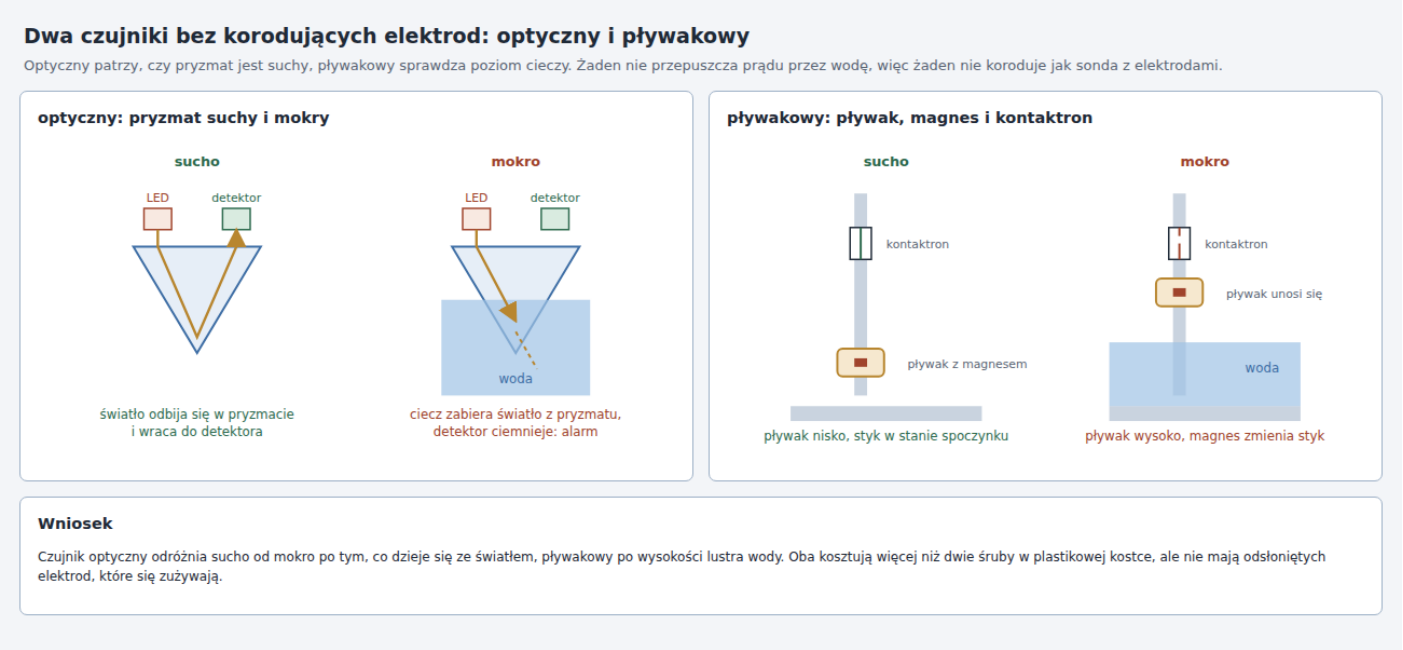
Zdjęcie do uzupełnienia: kabel sensoryczny ułożony w pętli pod klimatyzacją i wzdłuż ściany, z modułem lokalizującym na ścianie obok.

Zakres i dokładność: kable bywają dostępne w długościach od jednego do kilkuset metrów, a lokalizację podaje się z dokładnością rzędu metra, w najlepszych systemach nawet około 1 do 3 stóp, czyli około 0,3 do 0,9 metra [3][4]. Czas reakcji: sekundy od zamoczenia kabla. *Zalety: jedna linia pokrywa dużą powierzchnię, moduł pokazuje, gdzie szukać wycieku, kabel można przyłożyć do rury.* Wady: wyższy koszt, moduł lokalizujący trzeba zamawiać razem z kablem, a kable mają ograniczoną długość i liczbę połączeń. * Gdzie: serwerownie z podłogą techniczną, korytarze z rurami, pod sufitem podwieszanym, przy nawilżaczach.

Optyczny

> **Pojęcie: całkowite wewnętrzne odbicie.** Zjawisko, w którym światło biegnące w szkle albo przezroczystym tworzywie odbija się od powierzchni zamiast ją opuścić, jeśli po drugiej stronie jest powietrze. Gdy po drugiej stronie pojawi się ciecz, część światła wychodzi na zewnątrz.

W czujniku optycznym dioda świeci w pryzmat o kącie 45 stopni, a fotodioda patrzy na odbite światło. W powietrzu światło wraca do detektora, w wodzie wychodzi w ciecz i detektor ciemnieje [5]. Czujnik nie ma odsłoniętych elektrod, więc nie koroduje i nie zależy od przewodności cieczy.



Rysunek 3. Czujnik optyczny z pryzmatem oraz czujnik pływakowy. Oba nie mają elektrod, które się zużywają, a reagują na poziom cieczy. (kliknięcie otwiera pełny rozmiar)

Zakres i dokładność: wykrywa ciecz w punkcie przylegającym do pryzmatu; nie rozróżnia wody od oleju i innych cieczy. Czas reakcji: sekundy. *Zalety: bez elektrod, działa w cieczach, które nie przewodzą prądu, odporny na kurz.* Wady: wyższa cena niż czujnik przewodnościowy, reaguje także na krople rosy i kondensat na pryzmacie. * Gdzie: zbiorniki, tace na oleje i paliwa, przemysł, miejsca, gdzie wilgoć i chemikalia niszczyłyby elektrody.

Pływakowy

> **Pojęcie: czujnik pływakowy.** Pływak unosi się razem z poziomem wody i mechanicznie albo przez magnes przestawia styk. Poziom, przy którym styk się zmienia, ustawia konstrukcja czujnika.

Wyłącznik pływakowy w pozycji włączonej. Pływak unosi się z poziomem wody i przestawia styk, dla

Wyłącznik pływakowy w pozycji włączonej. Pływak unosi się z poziomem wody i przestawia styk, dlatego czujnik pływakowy zadziała dopiero wtedy, gdy woda sięgnie jego poziomu. Źródło: Wikimedia Commons (autor i wolna licencja podane na stronie pliku).

W wielu wykonaniach magnes w pływaku zamyka kontaktron umieszczony w trzonie, więc w środku nie ma przewodów odsłoniętych na wodę. Taki czujnik opisujemy dokładniej na stronie Kontaktrony i czujki ruchu: rodzaje, zasada działania i zastosowania, bo ten sam kontaktron pilnuje też drzwi.

Pionowy wyłącznik pływakowy w obudowie przemysłowej. Taki czujnik stawiamy w tacy albo studzie

Pionowy wyłącznik pływakowy w obudowie przemysłowej. Taki czujnik stawiamy w tacy albo studzience, gdzie woda może wzrosnąć o kilka centymetrów. Źródło: Wikimedia Commons (autor i wolna licencja podane na stronie pliku).

Zakres i dokładność: reaguje na poziom wody w punkcie, zwykle od kilkunastu milimetrów w górę, zależnie od konstrukcji. Czas reakcji: zależy od tempa przyrostu poziomu, bo pływak musi się unieść. *Zalety: prosty, nie ma elektrod, mechanizm działa bez zasilania czujnika.* Wady: potrzebuje głębokości, więc nie wykryje cienkiej warstwy na podłodze, a osad i kamień mogą go zablokować. * Gdzie: tace skroplin, studzienki, zbiorniki, pompownie.

Zawory odcinające

> **Pojęcie: zawór odcinający.** Zawór, który po sygnale z czujnika zamyka dopływ wody. W dużych instalacjach to zawór kulowy z siłownikiem elektrycznym, w małych zawór z cewką, czyli elektromagnetyczny.

Zawory elektromagnetyczne. Zamykają przepływ po sygnale elektrycznym, ale zamykają go szybko

Zawory elektromagnetyczne. Zamykają przepływ po sygnale elektrycznym, ale zamykają go szybko. Źródło: Wikimedia Commons (autor i wolna licencja podane na stronie pliku).

Zawór elektromagnetyczny zamyka się w ułamku sekundy, a nagłe zatrzymanie płynącej wody wywołuje uderzenie hydrauliczne, czyli falę ciśnienia, która potrafi uszkodzić rury i armaturę [6]. Dlatego przy większych średnicach stosuje się zawory kulowe z siłownikiem, które zamykają się wolniej [6]. W serwerowni Zielonej zaworu odcinającego nie potrzebujemy, bo mapa f0-02 wymienia tylko odpływ skroplin, a nie rury z wodą zasilającą. Przy klimatyzacji z obiegiem wody lodowej albo nawilżaczem parowym zawór staje się częścią projektu.

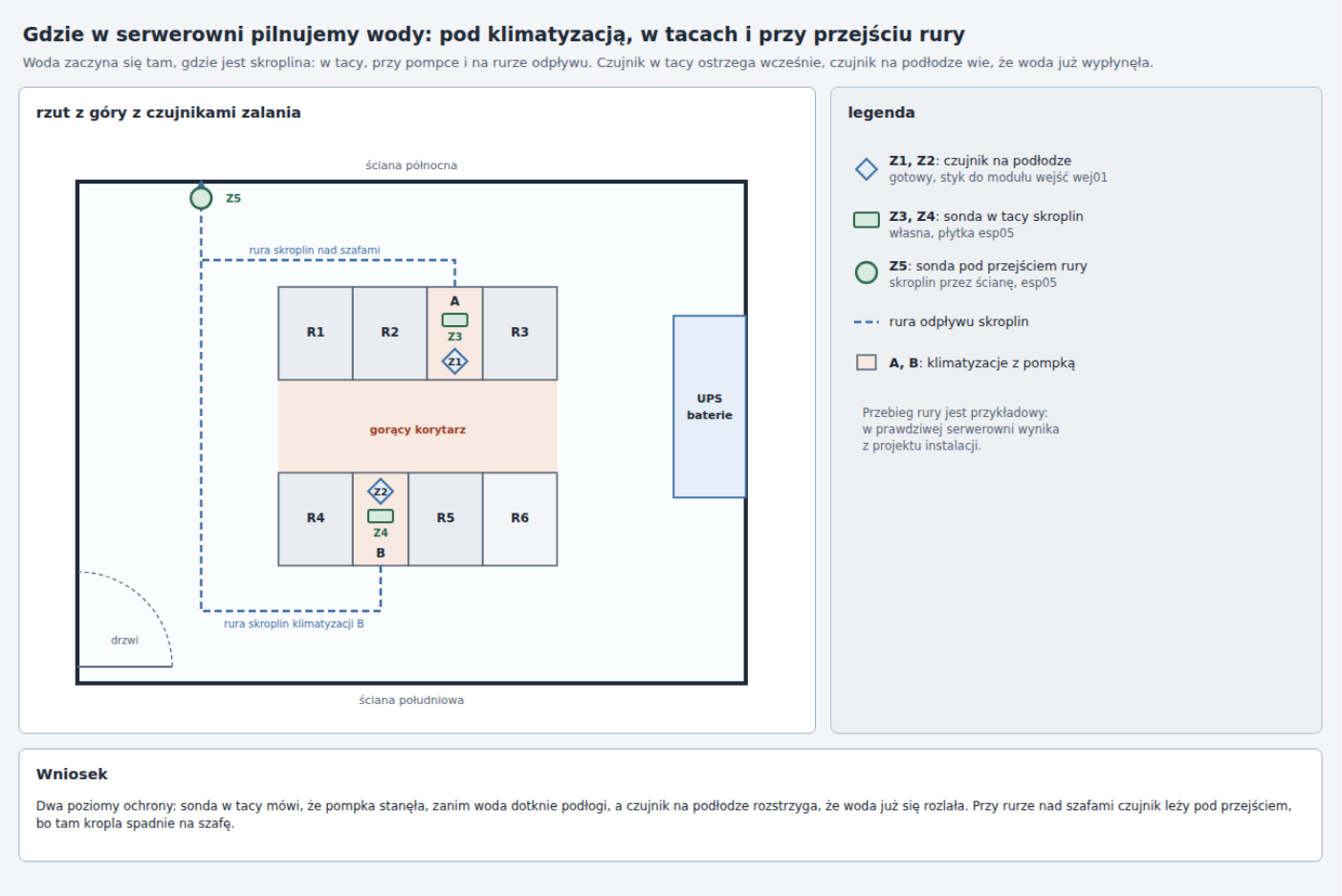
Nie zamykamy automatycznie dopływu wody do instalacji tryskaczowej ani zasilania instalacji przeciwpożarowej na podstawie samego czujnika zalania. To nasze zalecenie: taką decyzję podejmuje projektant z rzeczoznawcą do spraw zabezpieczeń przeciwpożarowych, bo fałszywy alarm wody wyłączyłby ochronę pożarową.

Wszystkie rodzaje w jednej tabeli

Rodzaj	Co wykrywa	Najlepszy przy	Najsłabszy przy	Typowe miejsce
Punktowy dwuelektrodowy	wodę między elektrodami	tacach, podłodze pod urządzeniem	wodzie o małej przewodności, kurzu	pod klimatyzatorem
Kabel sensoryczny	wodę wzdłuż kabla	dużych powierzchniach, rurach	krótkich pomieszczeniach, kosztach	podłoga techniczna, korytarz
Optyczny	ciecz przy pryzmacie	cieczach nieprzewodzących, zbiornikach	mgle i rosie na pryzmacie	zbiornik, taca na olej
Pływakowy	poziom wody	studzienkach, tacach skroplin	cienkiej warstwie wody	pod pompką, studzienka
Zawór odcinający	nie wykrywa, zamyka dopływ	nagłych wyciekach z rur	instalacji ppoż.	wejście rury do pomieszczenia

W serwerowni

Woda w serwerowni pojawia się tam, gdzie przepływa albo skrapla się: pod klimatyzacją, w tacy skroplin, przy pompce i przy rurze odpływu. Skropliny z parownika spływają do tacy, a stamtąd pompką do odpływu. Pompka jest małą, stale pracującą częścią z ruchomymi elementami, więc zawodzi najczęściej [4].



Rysunek 4. Rzut serwerowni z czujnikami zalania: Z1 i Z2 na podłodze pod klimatyzacjami, Z3 i Z4 w tacach skroplin, Z5 pod przejściem rury przez ścianę. Przebieg rury jest przykładowy i w prawdziwej serwerowni wynika z projektu instalacji. (kliknięcie otwiera pełny rozmiar)

Zalecenia, które stosujemy w Zielonej i zalecamy w innych pomieszczeniach:

Pod klimatyzacją: czujnik w tacy i czujnik na podłodze obok niej. Taca mówi, że pompka stanęła, a czujnik na podłodze rozstrzyga, że woda już wypłynęła [4]. Przy skroplinach: czujnik pod złączkami i pompką, bo to najczęstsze miejsce rozszczelnienia. *Pod przejściem rury przez ścianę albo sufit: czujnik w miejscu, w którym kropla spadnie na szafę, a nie tam, gdzie wygodniej go zamontować.* Nad sufitem podwieszanym i pod rurami nad szafami: kabel sensoryczny albo czujniki w tacach, jeśli rury z wodą przechodzą nad sprzętem. *Przy nawilżaczu i przy rurach wody lodowej, jeśli je mamy: czujnik pod każdym połączeniem i na armaturze [4].* W kilku warstwach: kilka czujników w różnych miejscach daje szybsze ostrzeżenie niż jeden na końcu drogi wody [4].

Czego nie robimy: nie stawiamy czujnika wyłącznie przy drzwiach, nie układamy kabla w miejscu, gdzie często chodzą ludzie, i nie zakładamy, że wykryty wyciek zawsze znaczy awarię klimatyzacji. Przy każdym wykryciu wody sprawdzamy też rury nad szafami.

Monitoring w Zabbixie

Czujniki Z1 i Z2 z mapy serwerowni to gotowe czujniki podłogowe ze stykiem, podłączone do modułu wejść `wej01`. Moduł wysyła przy wodzie własny SMS, niezależny od `mon01`, a Zabbix odczytuje jego stan przez SNMP i pokazuje historię. Szczegóły tej drogi opisuje strona Serwerownia Zielonej: gdzie stoją czujniki i dokąd płyną alarmy.

Czujniki Z3, Z4 i Z5 to nasze uzupełnienie: własna płytka z ESP32 pilnuje tac skroplin i przejścia rury, a Zabbix na `mon01` odpytuje ją co 30 sekund. Budowę opisuje strona Czujniki zalania w praktyce: wybór, montaż i własna sonda (DIY). Każdy z tych czujników ma też wyzwalacz braku danych, bo czujnik, który milczy, nie może wyglądać jak czujnik z suchą podłogą.

Gotowe urządzenia komercyjne

Nie polecamy konkretnych marek, ale kilka klas urządzeń warto znać:

Klasa	Co robi	Uwagi
Czujnik podłogowy ze stykiem i przekaźnikiem	zwiera albo rozwiera styk po wykryciu wody	idzie do modułu wejść, pożądany nadzór linii rezystorami
Kabel sensoryczny z modułem lokalizującym	wskazuje miejsce wycieku wzdłuż kabla	często z wyjściami przekaźnikowymi, SNMP albo Modbus
Czujnik bezprzewodowy na baterii	wysyła alarm radiowo do bramki	łatwy montaż, wymaga kontroli baterii i zasięgu
Moduł wejść z modemem GSM	zbiera styki i wysyła SMS bez serwera	nasz `wej01`
Zawór z siłownikiem i sterownikiem	zamyka dopływ po sygnale z czujnika	wymaga projektu i zgody właściciela instalacji

Poza serwerownią

W domu czujnik zalania stawiamy pod zmywarką, pralką, bojlerem i w łazience, a przy wejściu wody do mieszkania zawór z siłownikiem potrafi zamknąć dopływ, gdy czujnik zgłosi wyciek. Takie zestawy sprzedaje się jako gotowe urządzenia z baterią i radiem, a dla smarthome ważne jest, że czujnik zgłasza się do centrali, nie tylko piszczy. Przy zaworze liczy się ta sama uwaga o uderzeniu hydraulicznym [6].

W magazynie w Swarzędzu woda pojawia się tam, gdzie są rury i urządzenia z wodą: przy węźle, w kotłowni i pod urządzeniami chłodniczymi, a także pod dachem przy rynnach wewnętrznych. Stawiamy tam czujniki punktowe i, jeśli rura biegnie nad towarem, kabel pod rurą. Wyciek nad paletą z towarem spożywczym jest kosztowny, ale o rozmieszczeniu decyduje układ konkretnych rur, a nie ta strona.

W przemyśle czujniki wycieku pilnują zbiorników, rurociągów z cieczami i tac na paliwa, a optyczne czujniki poziomu działają w cieczach, których elektrody by nie wykryły [5].

Mity i sprostowania

Obiegowa wersja	Jak jest naprawdę
Czujnik zalania wykryje każdą wodę	Czujnik dwuelektrodowy potrzebuje przewodnictwa, a woda o małej zawartości soli, jak skropliny, przewodzi słabo, więc czułość trzeba dobrać i sprawdzić testem
Wystarczy jeden czujnik na podłodze	Woda z rury nad szafą spada na sprzęt, zanim dotrze do podłogi, a w tacy czujnik ostrzega wcześniej niż na podłodze [4]
Zawór elektromagnetyczny to najlepszy sposób zamknięcia wody	Szybkie zamknięcie powoduje uderzenie hydrauliczne, dlatego stosuje się zawory kulowe z siłownikiem, które zamykają się wolniej [6]
Czujnik zalania ma jednego znanego wynalazcę	Nie udało nam się tego ustalić: znamy składniki, czyli przewodnictwo, pływak i optykę, ale nie autora gotowego czujnika
Raz zamontowany czujnik działa bez końca	Elektrody korodują, kurz i osad zmieniają czułość, dlatego zwilżamy czujnik ściereczką raz na kwartał zgodnie z planem przeglądów
Czujnik z baterią nie wymaga uwagi	Bateria się wyczerpuje, a brak sygnału wygląda jak sucha podłoga, więc potrzebny jest nadzór łączności

Oś czasu

Data	Wydarzenie
około 250 p.n.e.	Ktezybiusz z Aleksandrii buduje zegar wodny z pływakowym regulatorem poziomu [1]
1800	Nicholson i Carlisle rozkładają wodę prądem na wodór i tlen [2]
1834	Faraday nazywa proces elektrolizą i podaje prawa elektrolizy [2]

Data	Wydarzenie
2017	Polska firma Fibaro przedstawia czujnik zalania z obsługą HomeKit, według prasy pierwszy taki czujnik [7]

Wątek polski

Z polskich wynalazców nie znaleźliśmy żadnego, który byłby autorem czujnika zalania. Nie znaleźliśmy też odrębnej polskiej normy, która określałaby czujniki zalania w serwerowni. Wymagania dla serwerowni wynikają z projektu, z zaleceń producentów klimatyzacji i z wymagań ubezpieczyciela.

Polski wątek jest natomiast produktowy. Polska firma Fibaro oferuje czujnik zalania Flood Sensor FGFS-101, który zgłasza się do centrali smarthome, a według prasy w 2017 roku był pierwszym czujnikiem zalania z obsługą HomeKit [7][8]. To przykład klasy czujnika radiowego na baterii, który w domu zastępuje alarm syreną, a w serwerowni nie zastąpi czujnika podłączonego kablem do modułu wejść z nadzorem linii.

Częste błędy

Objaw	Przyczyna	Co robimy
Alarm zalania bez wody	kurz, kondensat albo osad na elektrodach	czyścimy elektrody, sprawdzamy położenie czujnika, ewentualnie zmieniamy rodzaj czujnika
Cisza mimo wycieku	przerwany kabel, wyczerpana bateria albo czujnik wysoko nad podłogą	sprawdzamy nadzór linii, baterię i wysokość, robimy test ściereczką
Czujnik w tacy milczy, a woda jest na podłodze	taca przelała się dołem albo woda przeszła obok	dokładamy czujnik na podłodze, układamy czujniki w dwóch warstwach
Zawór po zamknięciu uszkodził rurę	uderzenie hydrauliczne po zbyt szybkim zamknięciu	stosujemy zawór kulowy z siłownikiem albo zawór o regulowanym czasie zamykania [6]
Kabel sensoryczny pokazuje wyciek w złym miejscu	wilgoć w połączeniu albo przerwa w kablu	sprawdzamy połączenia i długość kabla, porównujemy z dokumentacją
Alarm zalania wyłączył tryskacze	czujnik wpięty w sterowanie instalacją ppoż. bez projektu	odłączamy, takie połączenie robi tylko uprawniony projektant

□□ Ćwiczenia

1. Dlaczego czujnik zasilany stałym napięciem z czasem przestaje działać?

Odpowiedź

Stałe napięcie na elektrodach w wodzie powoduje elektrolizę i korozję elektrod [2]. Czujnik zasilany impulsowo, na ułamek procent czasu, zużywa się dużo wolniej.

1. Który czujnik wybieramy do tacy na olej i dlaczego?

Odpowiedź

Optyczny albo pływakowy. Olej nie przewodzi prądu, więc czujnik przewodnościowy go nie wykryje, a optyczny reaguje na obecność cieczy przy pryzmacie [5].

1. W serwerowni leżą dwa czujniki pod klimatyzacją: jeden w tacy i jeden na podłodze. Co mówi każdy?

Odpowiedź

Czujnik w tacy mówi, że pompka stanęła albo odpływ jest zatkany, zanim woda wypłynie. Czujnik na podłodze mówi, że woda już się rozlała i zbliża się do szaf.

1. Dlaczego nie zamykamy automatycznie zasilania tryskaczy czujnikiem zalania?

Odpowiedź

Fałszywy alarm wody wyłączyłby ochronę pożarową. Taką decyzję podejmuje projektant z rzeczoznawcą do spraw zabezpieczeń przeciwpożarowych, a nie czujnik zalania.

1. Co to jest uderzenie hydrauliczne i jak się go unikamy?

Odpowiedź

To fala ciśnienia po nagłym zatrzymaniu płynącej wody, która potrafi uszkodzić rury [6]. Unikamy jej przez zawory kulowe z siłownikiem, które zamykają się wolniej niż zawory z cewką.

1. Kabel sensoryczny pokazuje wyciek 40 metrów od początku kabla. Jak moduł to wie?

Odpowiedź

Moduł mierzy spadek napięcia na drodze do miejsca zwarcia wodą. Spadek jest proporcjonalny do długości kabla, więc moduł zamienia go na odległość [3].

Powiązane: Czujniki zalania w praktyce: wybór, montaż i własna sonda (DIY) · Serwerownia Zielonej: gdzie stoją czujniki i dokąd płyną alarmy · Klimatyzacja serwerowni: rodzaje, free cooling i czynniki chłodnicze · Higrometry: rodzaje, zasada działania i zastosowania · Kontaktrony i czujki ruchu: rodzaje, zasada działania i zastosowania

Źródła:

1. EBSCO, Ctesibius of Alexandria, opis zegara wodnego i regulatora pływakowego: https://www.ebsco.com/research-starters/history/ctesibius-alexandria/
2. Wikipedia, Electrolysis, historia elektrolizy: https://en.wikipedia.org/wiki/Electrolysis
3. Chemelex (nVent), karta katalogowa kabla sensorycznego TraceTek TT1000: https://cdn.chemelex.com/Product%20Documents/Data%20Sheets/RAYCHEMTraceTek-DS-H53870-TT1000-EN.pdf
4. Icon Process Controls, Leak Detection Strategies for Data Centers and Mission-Critical Facilities, rozmieszczenie czujników punktowych i kabli: https://iconprocon.com/blog_post/leak-detection-strategies-for-data-centers-and-mission-critical-facilities/
5. Patent US4286464, czujnik poziomu cieczy z pryzmatem i całkowitym wewnętrznym odbiciem: https://patents.google.com/patent/US4286464
6. Tameson, Water Hammer, uderzenie hydrauliczne i zawory: https://tameson.com/water-hammer.html
7. iMagazine, Polski smart home Fibaro w Apple Store online, 2017: https://imagazine.pl/2017/10/13/polski-smart-home-fibaro-w-apple-store-online
8. Instrukcja czujnika zalania Fibaro FGFS-101: https://www.montersi.pl/instrukcje/fibaro-fgfs101-instrukcja.pdf

Czujniki zalania w praktyce: wybór, montaż i własna sonda (DIY)

Poziom: średnio

Dotyczy: ESP32 (Olimex ESP32-POE), ESPHome 2026.6, Zabbix 7.0 LTS, Home Assistant, moduł wejść wej01

Czas: ~90 min

Budujemy moduł `esp05` z trzema własnymi sondami zalania: dwie w tacach skroplin pod klimatyzacjami (Z3 i Z4) i jedna pod przejściem rury skroplin przez ścianę (Z5). Pokazujemy też, jak podłączyć gotowy czujnik podłogowy ze stykiem do modułu `wej01`, jak zrobić alarm w Zabbixie na `mon01` i jak sprawdzić całość wylaniem wody. Komendy ESPHome wykonujemy na stacji administratora z Pythonem 3.12 albo nowszym.

Skróty użyte na tej stronie (19)

Skrót	Co znaczy
API	Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą
BMS	Building Management System: system zarządzania budynkiem, który zbiera odczyty z czujników (temperatura, wilgotność) i steruje klimatyzacją oraz wentylacją
ESP32	Tani mikrokontroler z Wi-Fi i Bluetooth firmy Espressif: malutki komputer, z którego budujemy własne czujniki sieciowe
ESPHome	Oprogramowanie, które zamienia ESP32 w gotowy czujnik sieciowy bez pisania kodu od zera, a odczyty udostępnia przez sieć
GPIO	General Purpose Input Output: uniwersalny pin mikrokontrolera, który można ustawić jako wejście albo wyjście
GSM	Global System for Mobile Communications: standard sieci komórkowej, z którego korzysta modem wysyłający SMS
HTTP	Hypertext Transfer Protocol: protokół stron WWW
IT	Information Technology: dział albo praca związana z komputerami i infrastrukturą firmy
JSON	JavaScript Object Notation: tekstowy format danych z klamrami i cudzysłowami

Skrót	Co znaczy
JSONPath	Zapis ścieżki, która wskazuje jedną wartość w dokumencie JSON, np. $.value to pole value na samym wierzchu
LAN8720	układ scalony (PHY) obsługujący Ethernet 10/100 w płytkach ESP32 z kablem sieciowym, np. Olimex ESP32-POE; w ESPHome zapisujemy go jako type: LAN8720
LTS	Long Term Support: wersja z długim wsparciem poprawkami
Modbus	Prosty protokół komunikacji przemysłowej, którym liczniki energii i moduły pomiarowe wysyłają odczyty
PoE	Power over Ethernet: zasilanie urządzenia tym samym kablem sieciowym, którym płyną dane
SMS	Short Message Service: krótka wiadomość tekstowa na telefon, która dociera także bez internetu
SNMP	Simple Network Management Protocol: protokół odczytu stanu urządzeń sieciowych
URL	Uniform Resource Locator: adres strony lub zasobu
USB	Universal Serial Bus: standard portów i urządzeń podłączanych do komputera, tu pendrive'y i inne nośniki wymienne
YAML	YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach

Fiszka

□□ ZMIENIA SYSTEM

Instalacja ESPHome: pipx install esphome

```
pipx install esphome
```

ESPHome buduje program dla płytki ESP32 z jednego pliku YAML i wgrywa go przez USB albo przez sieć. Konfigurację z tej strony sprawdziliśmy poleceniem `esphome config` na wersji 2026.6.5, a w nowszych wersjach to samo polecenie pokaże ewentualne zmiany składni.

□□ **ESPHome** = Oprogramowanie, które zamienia ESP32 w gotowy czujnik sieciowy bez pisania kodu od zera, a odczyty udostępnia przez sieć · **ESP32** = Tani mikrokontroler z Wi-Fi i Bluetooth firmy Espressif: malutki komputer, z którego budujemy własne czujniki sieciowe · **YAML** = YAML Ain't Markup Language: format plików konfiguracyjnych oparty na wcięciach · **USB** = Universal Serial Bus: standard portów i urządzeń podłączanych do komputera, tu pendrive'y i inne nośniki wymienne · **pipx** = instaluje narzedzia napisane w Pythonie, kazde we wlasnym, odizolowanym srodowisku, bez mieszania zaleznosci miedzy programami

TYLKO ODCZYT

Wygenerowanie klucza API: openssl rand -base64 32

```
openssl rand -base64 32
```

Klucz szyfrujący natywne API to 32 losowe bajty zapisane w base64. ESPHome odrzuca każdy inny tekst [1].

API = Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą · **ESPHome** = Oprogramowanie, które zamienia ESP32 w gotowy czujnik sieciowy bez pisania kodu od zera, a odczyty udostępnia przez sieć · **openssl** = zestaw narzędzi do certyfikatów i szyfrowania

TYLKO ODCZYT

Plik z hasłami: secrets.yaml

```
# secrets.yaml, obok pliku esp05-zalanie.yaml
api_klucz: "9qHsey7aZDtFkU2T+l8H/6tstil5eWgau2ImI0tZH18="
ota_haslo: "HasloOta"
web_uzytkownik: "nazwa_usera"
web_haslo: "HasloWebServera"
```

Klucz w przykładzie jest poprawny składniowo, ale publiczny, bo stoi na tej stronie. Na prawdziwej płytce używamy własnego klucza z poprzedniej fiszki, a pozostałe wartości to placeholdery do podmiany. Hasła trzymamy w tym pliku, a w pliku konfiguracyjnym odwołujemy się do nich przez `!secret`.

PLIK KONFIGURACYJNY

Plik konfiguracyjny ESPHome: esp05-zalanie.yaml

```
esphome:
  name: esp05
  friendly_name: Sondy zalania

esp32:
  board: esp32-poe
  framework:
    type: esp-idf
```

```
logger:

# Ethernet z PoE: ustawienia dla Olimex ESP32-POE (wariant ISO ma te same piny)
ethernet:
  type: LAN8720
  mdc_pin: GPIO23
  mdio_pin: GPIO18
  clk:
    pin: GPIO17
    mode: CLK_OUT
  phy_addr: 0
  power_pin: GPIO12
  manual_ip:
    static_ip: 10.20.10.72
    gateway: 10.20.10.1
    subnet: 255.255.255.0
    dns1: 10.20.10.11

# Bez klienta API (Home Assistant) płytka restartowałaby się co 15 minut,
# a Zabbix czyta dane przez web_server, więc wyłączamy ten restart.
api:
  encryption:
    key: !secret api_klucz
  reboot_timeout: 0s

ota:
  - platform: esphome
    password: !secret ota_haslo

web_server:
  port: 80
  auth:
    username: !secret web_uzytkownik
    password: !secret web_haslo

# Wspólne zasilanie sond: pin daje 3,3 V tylko na 50 ms co 5 s
output:
  - platform: gpio
```

```
    pin: GPIO33
    id: zasilanie_sond

# Liczniki kolejnych mokrych próbek: alarm dopiero przy dwóch z rzędu
globals:
  - id: mokre_z3
    type: int
    initial_value: "0"
  - id: mokre_z4
    type: int
    initial_value: "0"
  - id: mokre_z5
    type: int
    initial_value: "0"

interval:
  - interval: 5s
    then:
      - output.turn_on: zasilanie_sond
      - delay: 50ms
      - lambda: |-
          id(mokre_z3) = id(surowy_z3).state ? id(mokre_z3) + 1 : 0;
          id(mokre_z4) = id(surowy_z4).state ? id(mokre_z4) + 1 : 0;
          id(mokre_z5) = id(surowy_z5).state ? id(mokre_z5) + 1 : 0;
      - output.turn_off: zasilanie_sond

binary_sensor:
  # Odczyty surowe, tylko do użytku wewnętrznego. Sondy Z3 i Z4: rezystor 220 kΩ do masy.
  - platform: gpio
    id: surowy_z3
    internal: true
    pin: GPIO35
  - platform: gpio
    id: surowy_z4
    internal: true
    pin: GPIO36
  # Moduł z komparatorem: wyjście DO jest niskie, gdy mokro, więc odwracamy.
  - platform: gpio
```

```
    id: surowy_z5
    internal: true
    pin:
      number: GPIO39
      inverted: true

  # Stany dla Zabbixa i Home Assistanta
  - platform: template
    name: "woda_z3"
    device_class: moisture
    lambda: |-
      return id(mokre_z3) >= 2;
  - platform: template
    name: "woda_z4"
    device_class: moisture
    lambda: |-
      return id(mokre_z4) >= 2;
  - platform: template
    name: "woda_z5"
    device_class: moisture
    lambda: |-
      return id(mokre_z5) >= 2;

sensor:
  # Bicie serca: brak odczytów z tej pozycji oznacza martwą płytkę albo sieć.
  - platform: uptime
    name: "esp05_uptime"
    update_interval: 60s
```

Plik opisuje płytkę Olimex ESP32-POE z Ethernetem i zasilaniem PoE (ustawienia układu LAN8720 z dokumentacji ESPHome [2][3]), szyfrowane API, aktualizacje przez sieć, prosty serwer HTTP oraz trzy stany wody. Pin GPIO33 co 5 sekund na 50 milisekund zasila sondy, a po 50 milisekundach program odczytuje wejścia GPIO35, GPIO36 i GPIO39 i wyłącza zasilanie. Stan wody pojawia się dopiero przy dwóch kolejnych mokrych próbkach, czyli po około 10 sekundach, co wycina pojedyncze zakłócenia. Nazwy encji nie mają spacji ani polskich liter, bo wchodzą do adresu URL [4].

Sekcja `api` ma `reboot_timeout: 0s`, bo bez klienta API płytka restartowałaby się co 15 minut, a u nas dane pobiera Zabbix przez serwer HTTP [1]. W Home Assistancie dodajemy płytkę jako urządzenie ESPHome i wpisujemy ten sam klucz.

ESPHome = Oprogramowanie, które zamienia ESP32 w gotowy czujnik sieciowy bez pisania kodu od zera, a odczyty udostępnia przez sieć · **PoE** = Power over Ethernet: zasilanie urządzenia tym samym kablem sieciowym, którym płyną dane · **LAN8720** = układ scalony (PHY) obsługujący Ethernet 10/100 w płytkach ESP32 z kablem sieciowym, np. Olimex ESP32-POE; w ESPHome zapisujemy go jako type: LAN8720 · **API** = Application Programming Interface: sposób, w jaki programy rozmawiają ze sobą · **HTTP** = Hypertext Transfer Protocol: protokół stron WWW · **URL** = Uniform Resource Locator: adres strony lub zasobu · **id** = pokaż użytkownika, jego numer UID oraz grupy, do których należy

TYLKO ODCZYT

Sprawdzenie pliku: esphome config esp05-zalanie.yaml

```
esphome config esp05-zalanie.yaml
```

Polecenie wczytuje plik wraz z `secrets.yaml`, sprawdza składnię i niczego nie wgrywa. Kończy się komunikatem `Configuration is valid!`, a o pinie GPIO12 wypisuje ostrzeżenie o pinach rozruchowych, które dla płytki Olimex ESP32-POE jest znane, bo ten pin zasila układ Ethernetu [3].

esphome = ESPHome: buduje i wgrywa oprogramowanie na ESP32 z pliku YAML, pokazuje logi urządzenia

TYLKO ODCZYT

Wgranie programu przez USB: esphome run esp05-zalanie.yaml

```
esphome run esp05-zalanie.yaml
```

Przy pierwszym wgraniu płytkę podłączamy kablem USB do stacji administratora. Kolejne aktualizacje idą przez sieć, według sekcji `ota`.

USB = Universal Serial Bus: standard portów i urządzeń podłączanych do komputera, tu pendrive'y i inne nośniki wymienne · **esphome** = ESPHome: buduje i wgrywa oprogramowanie na ESP32 z pliku YAML, pokazuje logi urządzenia

TYLKO ODCZYT

Podgląd dziennika: esphome logs esp05-zalanie.yaml

```
esphome logs esp05-zalanie.yaml
```

W dzienniku widać start Ethernetu, adres płytki i zmiany stanu encji `woda_z3`, `woda_z4` i `woda_z5`.

esphome = ESPHome: buduje i wgrywa oprogramowanie na ESP32 z pliku YAML, pokazuje logi urządzenia

TYLKO ODCZYT

Odczyt przez REST: curl

```
curl --user nazwa_usera http://10.20.10.72/binary_sensor/woda_z3
```

Przy takim zapisie `curl` pyta o hasło, więc nie zostaje ono w historii poleceń. Odpowiedź to obiekt JSON z polem `value`, które Zabbix wyciąga w następnym kroku [4].

Przykład:

```
{"id":"binary_sensor/woda_z3","state":"OFF","value":false}
```

Dokładny zapis pola `id` zależy od wersji ESPHome, a Zabbix korzysta tylko z pola `value`.

REST = sposób budowania API przez zwykłe żądania HTTP (GET, POST) zamiast osobnego protokołu · **JSON** = JavaScript Object Notation: tekstowy format danych z klamrami i cudzysłowami · **ESPHome** = Oprogramowanie, które zamienia ESP32 w gotowy czujnik sieciowy bez pisania kodu od zera, a odczyty udostępnia przez sieć · **curl** = client URL: pobieranie i wysyłanie przez HTTP z terminala

TYLKO ODCZYT

Odczyt wejścia modułu wejść przez SNMP: snmpget

```
snmpget -v2c -c NazwaWspolnoty AdresWej01 OIDWejscia
```

(gdzie `NazwaWspolnoty` to ciąg wspólnoty SNMP ustawiony w `wej01`, `AdresWej01` to adres modułu z puli serwerowni, a `OIDWejscia` to identyfikator obiektu wejścia Z1 z dokumentacji modułu)

Odpowiedź zależy od modelu modułu, więc jej nie przytaczamy. Wartości identyfikatorów i znaczenie liczb sprawdzamy w dokumentacji `wej01`.

SNMP = Simple Network Management Protocol: protokół odczytu stanu urządzeń sieciowych · **snmpget** = SNMP get: odczytuje jedną wartość urządzenia przez SNMP po jej numerze OID

Z życia wzięte

> **Z ŻYCIA WZIĘTE**
>
> Poniedziałek, 10:15. Tomek stoi nad tacą pod klimatyzacją A z dwiema nierdzewnymi śrubami, plastikową płytką i zwojem skrętki. Przykręca sondę tak, że końcówki śrub są kilka milimetrów nad dnem, a w Zabbixie encja `woda_z3` pokazuje stan wyłączony. Wylewa pół szklanki wody na dno tacy i po kilkunastu sekundach stan zmienia się na włączony, a po wytarciu wraca do wyłączonego. Potem odpina kabel przy sondzie i widzi, że sonda milczy, jakby było sucho, więc dopisuje do planu przeglądów test kwartalny ściereczką.

Projekt DIY krok po kroku

Moduł `esp05` ma trzy sondy i jedno wspólne wyjście zasilające. Dlaczego akurat tacki i przejście rury, a nie cała podłoga, wyjaśnia strona Czujniki zalania i wycieku: rodzaje, zasada działania i zastosowania.

Lista części

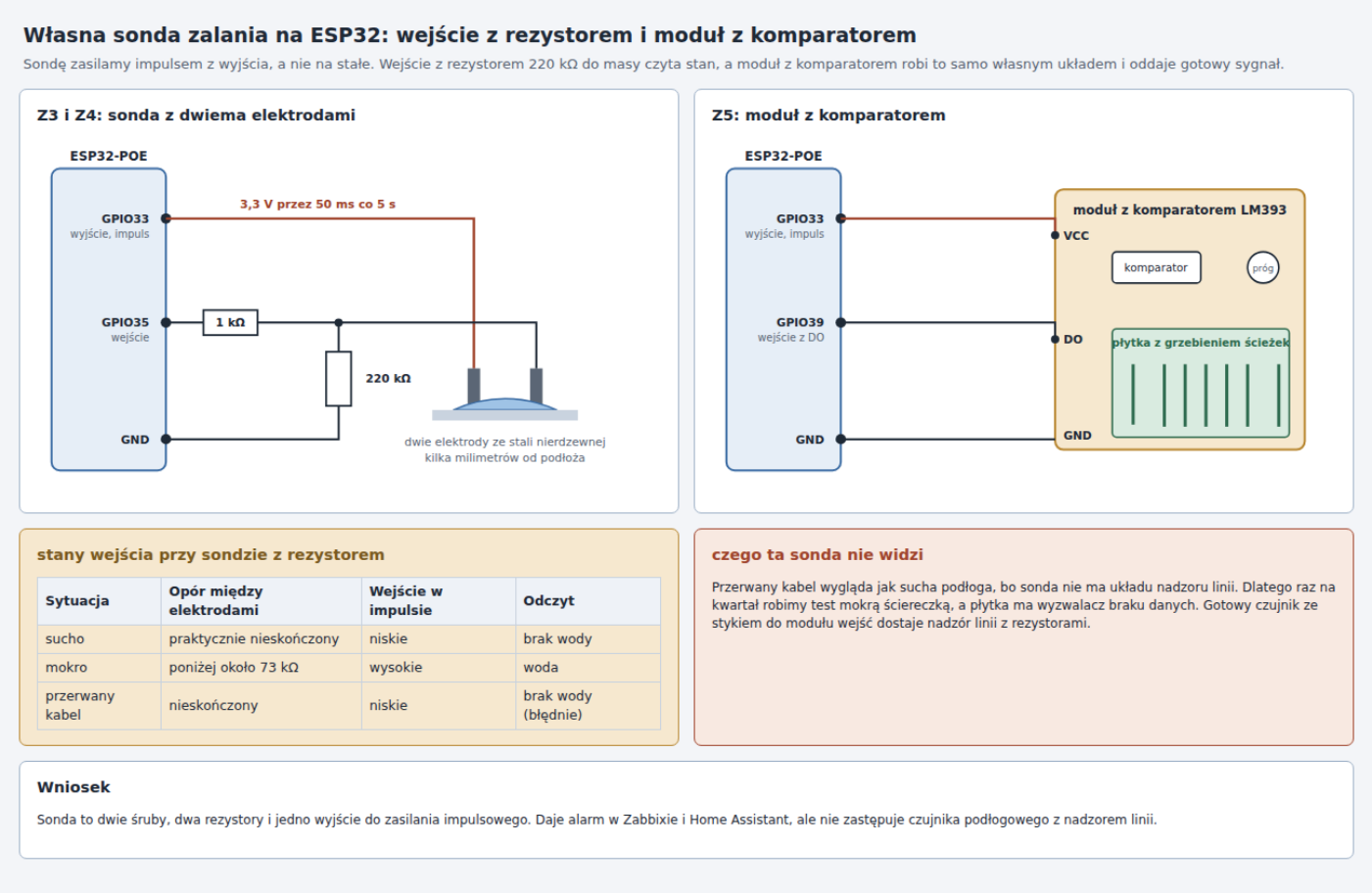
Część	Po co	Uwagi
Olimex ESP32-POE (albo ESP32-POE-ISO)	procesor z Ethernetem i zasilaniem z przełącznika	wariant ISO ma izolację galwaniczną
Cztery śruby ze stali nierdzewnej, na przykład M4, z nakrętkami	elektrody sond Z3 i Z4	stal nierdzewna koroduje wolniej niż miedź i stal zwykła
Dwie płytki z tworzywa i zaczepy do tacy	podstawa sondy	elektrody trzymamy w stałym rozstawie, kilka milimetrów nad dnem
Dwa rezystory 220 kΩ	opór do masy na wejściach sond	jeden na sondę
Dwa rezystory 1 kΩ	ochrona wejścia	jeden na sondę, w szeregu z wejściem
Moduł z komparatorem LM393 i płytką z grzebieniem ścieżek	sonda Z5, gotowy układ progu	wyjście cyfrowe DO jest niskie przy wodzie
Kabel skrętka kilkużyłowy	doprowadzenie do sond	krótkie odcinki, rząd kilku metrów
Kondensatory 10 nF, opcjonalnie	filtr zakłóceń na wejściu	kondensator między wejściem a masą
Obudowa na płytkę	ochrona przed kurzem i zwarciem	montaż powyżej poziomu podłogi
Port PoE w przełączniku (802.3af)	zasilanie i sieć jednym kablem	ten sam typ co przy modułach `esp01` i `esp02`

Część	Po co	Uwagi
Opaski, listwy i taśma dwustronna	mocowanie kabli i sond	z tworzywa

Płytka rozwojowa ESP32, poglądowa. Używamy Olimex ESP32-POE, który ma dodatkowo złącze Ethe

Płytka rozwojowa ESP32, poglądowa. Używamy Olimex ESP32-POE, który ma dodatkowo złącze Ethernet z zasilaniem PoE. Źródło: Wikimedia Commons (autor i wolna licencja podane na stronie pliku).

Schemat połączeń



Rysunek 1. Schemat sond: Z3 i Z4 to dwie elektrody z rezystorem 220 kΩ do masy i rezystorem 1 kΩ na wejściu, Z5 to moduł z komparatorem LM393 zasilany z tego samego pinu. Warto zauważyć, że oba układy są zasilane impulsem z jednego wyjścia GPIO33. (kliknięcie otwiera pełny rozmiar)

Wspólne wyjście GPIO33 daje 3,3 V przez 50 milisekund co 5 sekund, czyli przez około 1 procent czasu. Tak krótkie zasilanie ogranicza elektrolizę, o której piszemy na stronie teoretycznej, więc

elektrody zużywają się około stu razy wolniej niż przy stałym napięciu. Płynący prąd to najwyżej 15 mikroamperów, bo 3,3 V dzielimy przez 220 kΩ.

Wejście sondy łączymy przez rezystor 1 kΩ z GPIO35 albo GPIO36, a obok niego rezystor 220 kΩ do masy. Rezystor 1 kΩ chroni wejście przed przepięciami z kabla. Piny GPIO35, GPIO36 i GPIO39 są tylko wejściami i nie mają wewnętrznych rezystorów podciągających, więc ich nie konfigurujemy, a opór do masy dajemy zewnętrznie.

Próg wykrycia wynika z dzielnika. ESP32 uznaje za poziom wysoki napięcie od 0,75 napięcia zasilania, czyli około 2,5 V [5]. Żeby napięcie na rezystorze 220 kΩ osiągnęło ten poziom, opór wody między elektrodami musi być mniejszy niż jedna trzecia z 220 kΩ, czyli około 73 kΩ. Sucha podłoga i sucha taca mają opór praktycznie nieskończony, więc wejście jest niskie.

> **Pojęcie: dzielnik napięcia.** Dwa rezystory połączone szeregowo, między którymi odbieramy część napięcia. Napięcie na dolnym rezystorze to napięcie źródła razy opór dolnego podzielony przez sumę obu oporów.

Sonda Z5 używa gotowego modułu z komparatorem LM393: płytka z grzebieniem ścieżek, układ porównujący i potencjometr progu. Zasilamy go z GPIO33, wyjście DO prowadzimy na GPIO39, a przy wodzie wyjście schodzi do stanu niskiego, więc w pliku odwracamy je ustawieniem `inverted: true`. Moduł pobiera niewielki prąd, a GPIO ESP32 może oddać około 40 mA, ale pobór sprawdzamy w karcie modułu [5].

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zdjęcie do zrobienia: gotowa sonda Z3 w tacy skroplin, z dwiema nierdzewnymi śrubami na płytce z tworzywa i przepiętym kablem.

Budowa sondy

Sondę Z3 i Z4 budujemy z płytki z tworzywa, w której osadzamy dwie nierdzewne śruby tak, żeby ich końcówki wystawały po jednej stronie. Rozstaw końcówek dobieramy doświadczalnie, zaczynając od około 10 milimetrów, a wysokość nad dnem tacy od kilku milimetrów. Im mniejszy rozstaw i niższe położenie, tym wcześniej sonda reaguje, ale tym łatwiej o fałszywy alarm od kropli kondensatu. To nasze wymiary startowe, a nie norma, więc potwierdzamy je testem.

Śruby łączymy z kablem przez oczka lutowane i koszulki, a miejsce połączenia zabezpieczamy klejem lub koszulką, żeby woda nie dostała się pod izolację. Przewód zasilający idzie do jednej śruby, a przewód do wejścia przez rezystor 1 kΩ do drugiej. Długość kabla utrzymujemy w granicach kilku metrów, bo wejście o oporze 220 kΩ zbiera zakłócenia, a przy dłuższym kablu dodajemy kondensator 10 nF i sprawdzamy, czy odczyt nie skacze.

Pierwsze uruchomienie

Najpierw wgrywamy plik, podłączamy płytkę kablem Ethernet do portu PoE i w dzienniku sprawdzamy, że Ethernet się uruchomił i płytka ma adres 10.20.10.72. Potem odczytujemy `woda_z3`, `woda_z4` i `woda_z5` poleceniem `curl` i widzimy trzy wartości fałszywe. Dopiero wtedy mocujemy sondy w tacach i pod przejściem rury.

Kalibracja i test wylania

Test robimy w kontrolowany sposób, z wiadrem i ściereczką, nigdy przy włączonym sprzęcie obok. Najpierw przy suchej sondzie odczytujemy stan z `curl` i sprawdzamy, że jest wyłączony. Potem wlewamy około 50 mililitrów wody z kranu do tacy, tak żeby objęła obie elektrody, i mierzymy czas do zmiany stanu na włączony, który powinien wynosić około 10 sekund.

Następnie powtarzamy próbę z wodą pobraną ze skroplin klimatyzacji. Woda o małej zawartości soli przewodzi gorzej niż woda z kranu, więc to ten test mówi, czy próg 73 kΩ wystarcza. Jeśli sonda nie reaguje, zmniejszamy rozstaw elektrod albo zwiększamy rezystor do masy, a po każdej zmianie testujemy ponownie.

Wyniki zapisujemy w tabeli, którą dopisujemy do dokumentacji serwerowni:

Sonda	Woda z kranu	Skropliny	Czas do alarmu	Uwagi
Z3	do uzupełnienia	do uzupełnienia	do uzupełnienia	po teście
Z4	do uzupełnienia	do uzupełnienia	do uzupełnienia	po teście
Z5	do uzupełnienia	do uzupełnienia	do uzupełnienia	po teście

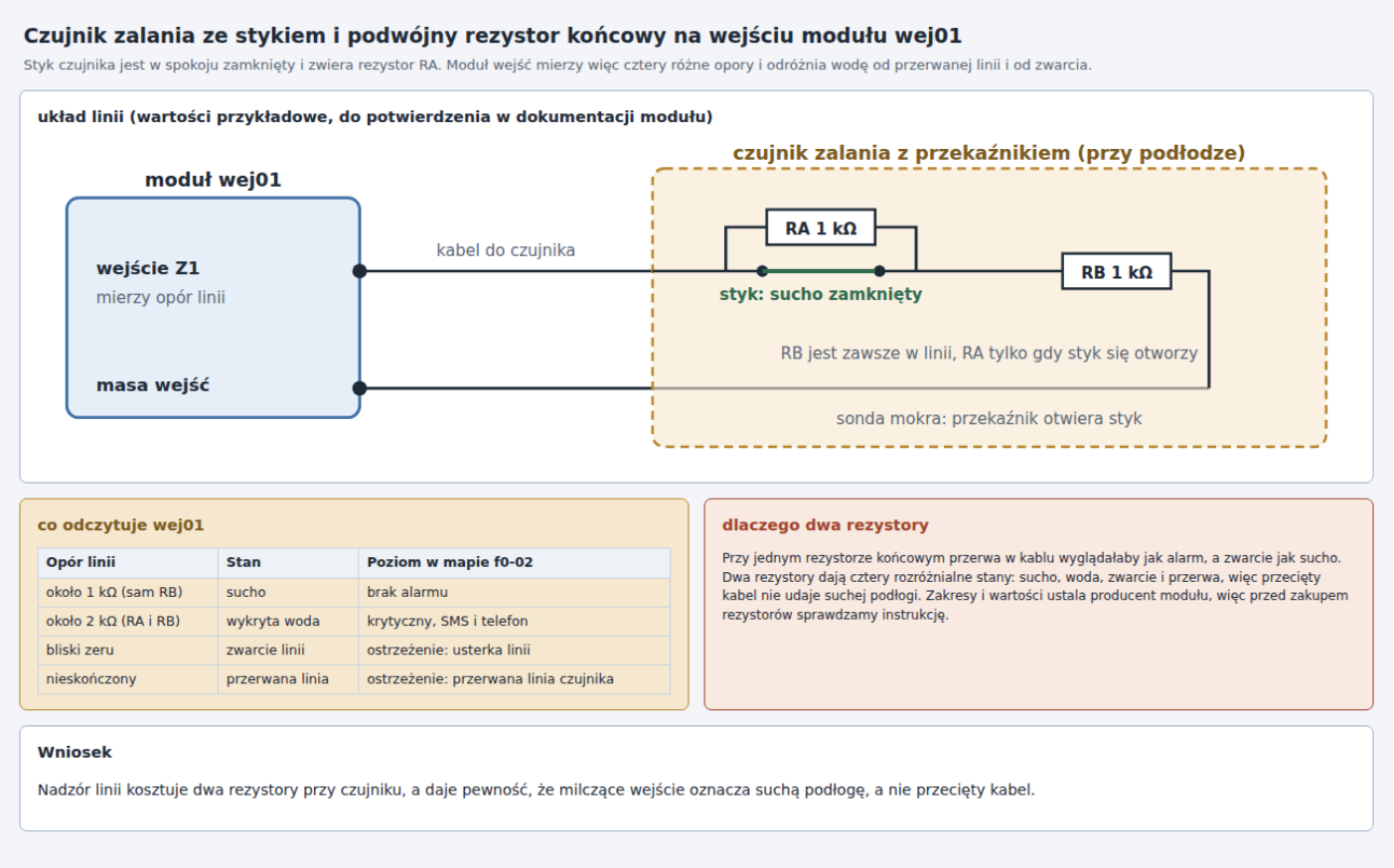
Na koniec odpinamy kabel przy sondzie i sprawdzamy, że stan zostaje wyłączony. To nie błąd programu, tylko ograniczenie tej konstrukcji: przerwany kabel wygląda jak sucha podłoga, o czym piszemy w sekcji o ograniczeniach.

Obudowa i miejsce montażu

Płytkę umieszczamy w obudowie na ścianie, powyżej poziomu podłogi i z dala od klimatyzacji, żeby nie dosięgła jej kałuża ani para z nawilżacza. Kable sond wychodzą z obudowy przez dławiki od dołu, żeby woda spływająca po kablu nie wpadała do środka. Sondy Z3 i Z4 przykręcamy do tac skroplin pod klimatyzacjami A i B, a Z5 układamy na podłodze pod przejściem rury skroplin przez ścianę, w miejscu, gdzie kropla spadnie pierwsza.

Gotowy czujnik podłogowy do wej01

Czujniki Z1 i Z2 z mapy serwerowni to gotowe czujniki ze stykiem, a nie nasze płytki. Ich styk podłączamy do wejścia modułu `wej01`, który sam wysyła SMS przy wodzie. W przeciwieństwie do naszej sondy taki czujnik może mieć nadzór linii: dwa rezystory przy czujniku pozwalają modułowi odróżnić wodę od przerwanego kabla.



Rysunek 2. Czujnik z przekaźnikiem i podwójnym rezystorem końcowym na wejściu modułu wejść: opór linii mówi, czy jest sucho, czy mokro, czy kabel jest przerwany albo zwarty. Wartości rezystorów są przykładowe i zależą od modułu. (kliknięcie otwiera pełny rozmiar)

Mapa serwerowni rozróżnia dwa stany: przerwaną linię czujnika jako ostrzeżenie i wykrytą wodę jako stan krytyczny. Zakładamy więc, że wejście `wej01` pracuje w układzie z dwoma rezystorami, a wartości i podłączenie podaje instrukcja modułu. To założenie wymaga potwierdzenia w dokumentacji konkretnego urządzenia, bo nie mamy jej w bazie.

W praktyce robimy trzy kroki. Najpierw sprawdzamy w instrukcji modułu, jakie rezystory końcowe przewiduje producent. Potem lutujemy je przy czujniku zgodnie z rysunkiem i w Zabbixie dodajemy

pozycje SNMP z fiszki, korzystając z identyfikatorów z dokumentacji. Na koniec testujemy ściereczką, rozwarciem kabla i zwarciem kabla, bo każdy z tych stanów powinien dać inny odczyt.

Drugi wariant: styk z esp05 do wolnego wejścia wej01

Alarm z naszej sondy może też iść do `wej01`, jeśli moduł ma wolne wejście. Wtedy SMS wychodzi z modemu GSM modułu, a nie z Zabbixa, więc działa nawet przy awarii `mon01`. Płytka nie ma styku, więc wyjście GPIO14 steruje tranzystorem i przekaźnikiem kontaktronowym, którego styk idzie do `wej01`.

W stanie spokoju cewka jest zasilona, a styk zamknięty. Przy wodzie albo przy braku zasilania płytki styk się otwiera i `wej01` zgłasza alarm. Dzięki temu awaria płytki i przerwa w kablu dają alarm, a nie ciszę. Plik rozszerzamy o wyjście i wspólny stan wody:

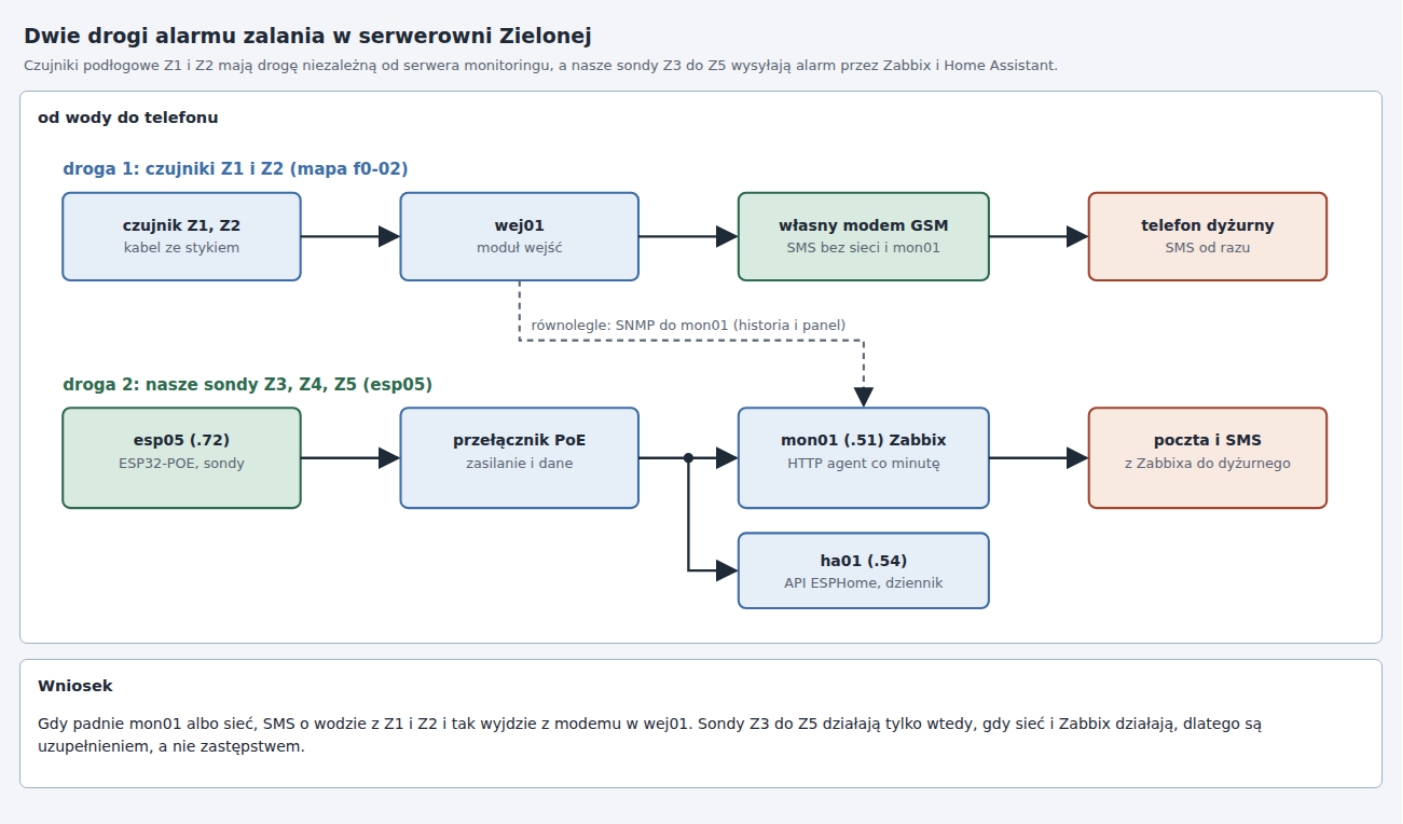
```
output:
  - platform: gpio
    pin: GPIO33
    id: zasilanie_sond
  - platform: gpio
    pin: GPIO14
    id: cewka_styku_wej01

binary_sensor:
  - platform: template
    name: "woda_dowolna"
    id: woda_dowolna
    device_class: moisture
    lambda: |-
      return id(mokre_z3) >= 2 || id(mokre_z4) >= 2 || id(mokre_z5) >= 2;
    on_press:
      - output.turn_off: cewka_styku_wej01
    on_release:
      - output.turn_on: cewka_styku_wej01
```

Fragment dopisujemy do pliku z fiszki: wyjście `cewka_styku_wej01` obok istniejącego `zasilanie_sond` i czujnik `woda_dowolna` do sekcji `binary_sensor`. Składnię sprawdziliśmy poleceniem `esphome config`, a działanie przekaźnika po starcie płytki i reakcję `wej01` sprawdzamy na urządzeniu. Czas reakcji `wej01` na zmianę wejścia sprawdzamy w dokumentacji, bo moduły często dodają opóźnienie.

◫ ZRZUT EKRANU DO UZUPEŁNIENIA

Zrzut do uzupełnienia: ustawienia wejścia modułu `wej01` z typem wejścia, czasem opóźnienia i opisem alarmu.

Integracja z monitoringiem



◫ *Rysunek 3. Dwie drogi alarmu zalania: czujniki Z1 i Z2 przez `wej01` do własnego modemu GSM i SMS-a, a nasze sondy Z3, Z4 i Z5 przez `esp05` do Zabbixa na `mon01` i Home Assistanta. Warto zauważyć, że droga przez `wej01` działa nawet wtedy, gdy `mon01` jest wyłączony. (kliknięcie otwiera pełny rozmiar)*

Zabbix co 30 sekund pyta płytkę o wartości przez serwer HTTP ESPHome, a odpowiedź JSON ma pole `value`, które element wyciąga ścieżką JSONPath [4][6]. Dzięki temu na płytce nie trzeba dodatków, a gdy milknie, Zabbix to widzi. Home Assistant dostaje te same dane przez natywne API i służy do podglądu i dziennika zdarzeń, a dodanie płytki opisuje strona ESPHome i urządzenia: adopcja esp01 i esp02, API albo MQTT, nazwy encji.

Host i makra w Zabbixie

Ścieżka w interfejsie, w której tworzymy host:

```
Data collection → Hosts → Create host
```

Host nazywamy `esp05`, dodajemy go do grupy `Czujniki środowiska` i nie dodajemy interfejsu agenta. Na hoście ustawiamy makra:

Makro	Wartość	Po co
`{$ESP.ADRES}`	`10.20.10.72`	adres płytki
`{$ESP.UZYTKOWNIK}`	nazwa użytkownika z sekcji `web_server`	uwierzytelnianie
`{$ESP.HASLO}`	hasło z sekcji `web_server`, typ Secret text	niewidoczne w interfejsie

Elementy danych

Wszystkie elementy mają typ HTTP agent, uwierzytelnianie podstawowe z makrami, interwał 30s i przetwarzanie wstępne JSONPath `$.value`. Dla stanów wody dodajemy jeszcze krok Boolean to decimal, który zamienia odpowiedź true i false na 1 i 0; dostępność tego kroku i rozpoznawane wartości sprawdzamy w dokumentacji naszej wersji Zabbixa [6].

Nazwa	Klucz	URL	Typ informacji
Woda, taca A (Z3)	`woda.z3`	`http://{$ESP.ADRES}/binary_sensor/woda_z3`	liczba całkowita bez znaku
Woda, taca B (Z4)	`woda.z4`	`http://{$ESP.ADRES}/binary_sensor/woda_z4`	liczba całkowita bez znaku
Woda pod przejściem rury (Z5)	`woda.z5`	`http://{$ESP.ADRES}/binary_sensor/woda_z5`	liczba całkowita bez znaku
Czas pracy płytki	`esp05.uptime`	`http://{$ESP.ADRES}/sensor/esp05_uptime`	liczba zmiennoprzecinkowa

Wyzwalacze

Wyrażenia zapisujemy w składni Zabbixa 7.0, w której nazwę hosta i klucz podajemy w ukośnikach [7]. Wykryta woda w tacy A, w tacy B i pod przejściem rury (Disaster, od razu SMS i telefon dyżurny jak dla Z1 i Z2):

```
last(/esp05/woda.z3)=1
last(/esp05/woda.z4)=1
last(/esp05/woda.z5)=1
```

Brak danych z płytki (High), bo czujnik, który milczy, nie może wyglądać jak czujnik z suchą podłogą:

```
nodata(/esp05/esp05.uptime,3m)=1
```

Restart płytki (Information), który pokazuje, że płytka wstała po awarii zasilania albo po aktualizacji:

```
last(/esp05/esp05.uptime)<300
```

Poziomy i czasy są naszymi propozycjami roboczymi i dostrajamy je po kilku tygodniach obserwacji. Odpowiadają mapie serwerowni, w której wykryta woda to stan krytyczny, a brak danych to ostrzeżenie dla zespołu IT.

Akcja: poczta i SMS

Ścieżka w interfejsie:

```
Alerts → Actions → Trigger actions → Create action
```

Dla poziomów od Warning w górę wysyłamy pocztę do grupy IT, a dla Disaster dodajemy SMS przez bramkę. Test wykonujemy bez wylewania wody: tymczasowo przestawiamy wyzwalacz na wartość 0, sprawdzamy, że alarm dociera, i przywracamy 1. Test z prawdziwą wodą robimy raz na kwartał ściereczką, zgodnie z planem przeglądów w mapie serwerowni.

Bezpieczeństwo i ograniczenia

Własna sonda jest uzupełnieniem monitoringu, a nie jego zastępstwem. Nie ma świadectwa wzorcowania ani certyfikatu, a jej czułość wynika z naszych pomiarów.

- Brak nadzoru linii: przerwany kabel albo odłączona elektroda wyglądają jak sucha podłoga, bo wejście nie odróżnia ich od braku wody. Łagodzimy to testem ściereczką raz na kwartał i wyzwalaczem braku danych, który zgłasza martwą płytkę, ale nie przerwany kabel.
- Kondensat i kurz: może dawać fałszywe alarmy, dlatego sondy czyścimy podczas przeglądu i stawiamy kilka milimetrów nad dnem.
- Napięcie: sondy pracują przy 3,3 V i prądzie rzędu mikroamperów, ale płytka stoi blisko urządzeń z napięciem sieciowym, więc prace wykonujemy przy wyłączonym sprzęcie i nie podłączamy sond do niczego innego niż wejścia płytki.
- Gwarancja i ubezpieczenie: urządzenie własnej roboty w serwerowni bywa sprzeczne z warunkami gwarancji sprzętu albo polisy, więc uzgadniamy je z zarządcą serwerowni i ubezpieczycielem.

- Serwer HTTP ESPHome domyślnie używa uwierzytelniania podstawowego [4]. Na `fw01` ograniczamy dostęp do portu 80 płytki do adresu `mon01` i nie wystawiamy go poza sieć serwerów.
- PoE: port przełącznika ma budżet mocy. Płytka z trzema sondami pobiera niewiele, ale sprawdzamy ten budżet przy podłączeniu.
- Aktualizacje: płytkę aktualizujemy przez sieć według sekcji `ota`, poza oknem zmian w listopadzie i styczniu.
- Sondy Z3 do Z5 nie zastępują czujników Z1 i Z2 na podłodze: tamte mają własny modem GSM i działają, gdy `mon01` jest wyłączony.

> Więcej możliwości albo gotowe rozwiązanie od ręki (out of the box): moduł wejść z modemem GSM i czujnikami z nadzorem linii, czujniki bezprzewodowe na baterii z bramką i integracją z Zabbixem, albo kabel sensoryczny z modułem lokalizującym podłączonym do sieci przez SNMP lub Modbus. W dużych centrach danych stosuje się kable lokalizujące pod całą podłogą techniczną, czujniki w każdej tacy i przy każdej rurze oraz połączenie z systemem BMS i zaworami odcinającymi z siłownikiem.

Częste błędy

Objaw	Przyczyna	Co robimy
`esphome config` zgłasza błąd klucza API	klucz nie jest 32 bajtami w base64	generujemy klucz poleceniem `openssl rand -base64 32`
`esphome config` ostrzega o pinie GPIO12	pin zasila Ethernet płytki Olimex	ostrzeżenie jest znane, ustawienia kopiujemy z dokumentacji [2][3]
Sonda cały czas pokazuje wodę	rezystor do masy nie jest podłączony albo kondensat łączy elektrody	sprawdzamy rezystor 220 kΩ, osuszamy elektrody, zwiększamy rozstaw
Sonda nie reaguje na wodę ze skroplin	opór wody przekracza próg około 73 kΩ	zmniejszamy rozstaw elektrod albo zwiększamy rezystor do masy
Stan zmienia się w nocy bez wody	zakłócenia na długim kablu	skracamy kabel, dodajemy kondensator 10 nF, skręcamy przewody
Moduł Z5 pokazuje odwrotnie	wyjście DO jest niskie przy wodzie	w pliku mamy `inverted: true` na wejściu GPIO39
Element w Zabbixie niewspierany	zły adres URL, hasło albo brak JSONPath	sprawdzamy `curl` z tego samego adresu i krok przetwarzania
Płytka restartuje się co 15 minut	`api` bez klienta i bez `reboot_timeout: 0s`	dopisujemy `reboot_timeout: 0s`

□□ Ćwiczenia

1. Dlaczego zasilamy sondy impulsem, a nie na stałe?

Odpowiedź

Stałe napięcie na elektrodach w wodzie powoduje elektrolizę, więc elektrody korodują. Zasilanie przez 50 milisekund co 5 sekund, czyli przez około 1 procent czasu, wydłuża ich życie.

1. Sonda ma rezystor 220 kΩ do masy, a próg wysokiego poziomu to 0,75 napięcia zasilania. Przy jakim największym oporze wody sonda zgłosi wodę?

Odpowiedź

Napięcie na rezystorze 220 kΩ musi wynosić co najmniej 0,75 napięcia źródła, więc opór wody nie może przekroczyć jednej trzeciej z 220 kΩ, czyli około 73 kΩ [5].

1. Odpinamy kabel przy sondzie i stan zostaje wyłączony. Czy płytka zgłosi problem?

Odpowiedź

Nie. Przerwany kabel wygląda jak sucha podłoga, bo konstrukcja nie ma nadzoru linii. Dlatego robimy test ściereczką raz na kwartał, a czujniki Z1 i Z2 z nadzorem linii pilnują podłogi niezależnie.

1. Po co dwa kolejne mokre odczyty, zanim stan zmieni się na włączony?

Odpowiedź

Pojedynczy impuls zakłócenia nie wywoła alarmu. Przy próbkowaniu co 5 sekund oznacza to opóźnienie rzędu 10 sekund, które uznajemy za rozsądne dla wody.

1. Dlaczego wyjście ze stykiem do `wej01` ma zasilaną cewkę w stanie spokoju?

Odpowiedź

Wtedy brak zasilania płytki albo przerwa w kablu otwiera styk i `wej01` zgłasza alarm. Układ jest bezpieczny przy awarii, bo awaria wygląda jak alarm, a nie jak cisza.

1. Dlaczego alarm z Z1 i Z2 wysyła SMS `wej01`, a nie Zabbix?

Odpowiedź

Gdy awaria wyłączy klaster z `mon01`, Zabbix i poczta znikają razem z serwerami. Moduł `wej01` ma własny modem GSM i wysyła SMS niezależnie od nich.

Powiązane: Czujniki zalania i wycieku: rodzaje, zasada działania i zastosowania · Serwerownia Zielonej: gdzie stoją czujniki i dokąd płyną alarmy · ESPHome i urządzenia: adopcja esp01 i esp02, API albo MQTT, nazwy encji · Klimatyzacja serwerowni: rodzaje, free cooling i czynniki chłodnicze

Źródła:

1. ESPHome, Native API Component: https://esphome.io/components/api/
2. ESPHome, Ethernet Component: https://esphome.io/components/ethernet/
3. Olimex, ESP32-POE na GitHub: https://github.com/OLIMEX/ESP32-POE
4. ESPHome, Web Server Component oraz Web Server API: https://esphome.io/components/web_server/ oraz https://esphome.io/web-api/
5. Espressif, ESP32 Datasheet, charakterystyka prądu stałego wejść i wyjść: https://documentation.espressif.com/esp32_datasheet_en.pdf
6. Zabbix 7.0, element HTTP agent: https://www.zabbix.com/documentation/7.0/en/manual/config/items/itemtypes/http
7. Zabbix 7.0, wyrażenia wyzwalaczy: https://www.zabbix.com/documentation/7.0/en/manual/config/triggers/expression