



AKADEMIA GÓRNICZO-HUTNICZA

Raport z projektu

„Urządzenie IoT monitorujące temperaturę i wilgotność w pomieszczeniu serwerowym”

z przedmiotu

Sensory w Aplikacjach Wbudowanych

Elektronika i Telekomunikacja

Systemy Wbudowane, rok I

Jakub Górnisiewicz

Przemysław Kapała

Marcin Maj

Jakub Zimnol

26.06.2023

Spis treści

1. OPIS I ZAŁOŻENIA PROJEKTU	3
Krótki opis projektu	3
Główny cel.....	3
Cel części softwareowej	3
Cel części hardwarowej	3
Przydział odpowiedzialności	4
2. SPRZĘT	5
Wybór komponentów	5
Opis schematu	5
Schematy	6
Symulacje.....	7
Layout.....	10
3. OPROGRAMOWANIE.....	10
4. URUCHOMIENIE.....	12
Problemy i komplikacje.....	12
Część sprzętowa	12
Część oprogramowania.....	14
Weryfikacja	15
Bibliografia	17

1. OPIS I ZAŁOŻENIA PROJEKTU

Krótki opis projektu

Główny cel

Celem projektu było zaprojektowanie urządzenia IoT (ang. *Internet of Things*), które monitoruje wartość temperatury i wilgotności w pomieszczeniu serwerowym oraz wysyła dane do serwera IoT.

W miejscach takich jak rozdzielnie czy serwerownie pomiary wilgotności i temperatury są kluczowe, gdyż mają one bezpośredni wpływ na działanie oraz żywotność sprzętu elektronicznego, a co za tym idzie również wpływa to znacząco na kwestie finansowe oraz ekologiczne. Niepożądanym jest, aby sprzęt był wymieniany zbyt często/nadmiarowo tylko i wyłącznie z powodu złej konserwacji urządzeń oraz niezapewnienia odpowiedniego środowiska pracy. Dodatkowo w serwerowniach z założenia znajduje się wiele przewodów, przez co jednym z założeń jest wyeliminowanie zbędnego okablowania.

Projekt podzielono na dwie części: oprogramowania (ang. *software*) oraz sprzętu (ang. *hardware*).

Cel części softwareowej

Celem części softwareowej było stworzenie programu składającego się z elementów typu open-source na mikrokontroler, którego celem jest odczytywanie wartości z czujnika temperatury/wilgotności i (periodyczne) przesyłanie tych danych do serwera IoT. Biorąc pod uwagę częsty brak dostępności niektórych zestawów rozwojowych z mikrokontrolerami oraz samych mikrokontrolerów, założono iż stworzone oprogramowanie powinno być łatwo implementowalne na różne architektury procesorów dostępnych komercyjnie na rynku.

Pozostałymi założeniami przy tworzeniu oprogramowania było:

- wykorzystanie systemu operacyjnego czasu rzeczywistego Zephyr OS w celu ułatwienia spełnienia założeń projektowych
- wykorzystanie protokołu LwM2M jako rozwiązania problemu telemetrii oraz zdalnego zarządzania urządzeniem (ang. *device management*)
- wykorzystanie istniejącego szkieletu programu klienta LwM2M Anjay oraz możliwości serwera Coiot DM
- generyczne podejście do tematu odczytywania danych z sensora w przypadku potrzeby zmiany posiadanego sensora na inny model/element

Cel części hardwarowej

Celem części sprzętowej było zaprojektowanie płytki PCB (ang. *Printed Circuit Board*) w programie KiCad. Płytką powinna umożliwiać połączenie sensora, mikrokontrolera oraz pozostałych wykorzystywanych w projekcie elementów elektronicznych. Komponenty użyte w projekcie muszą być dostępne na polskim rynku, tanie oraz być montowane w łatwy sposób.

Głównymi założeniami przy projektowaniu płytki było:

- odpowiednie rozmieszczenie elementów elektronicznych wykorzystanych w projekcie
- możliwość wykorzystania zasilania dostarczonego za pomocą kabla Ethernetowego z przełącznika sieciowego (ang. Switch) poprzez zastosowanie protokołu PoE (Power over Ethernet - IEEE 802.3af), a co za tym idzie wyeliminowanie konieczności użycia dodatkowego zasilacza
- odbiór sygnału wysłanego przez przełącznik sieciowy na porty SPI mikrokontrolera (ang. Serial Peripheral Interface)
- przetworzenie napięcia do odpowiedniej wartości, umożliwiającej poprawne działanie wykorzystywanych podzespołów i elementów
- dodatkowym aspektem było zaprojektowanie płytki w taki sposób, aby można było w łatwy sposób zweryfikować poprawność działania segmentów (zastosowanie zworek)

Przydział odpowiedzialności

Tabela 1. Przydział obowiązków

	Jakub G.	Przemysław K.	Marcin M.	Jakub Z.
Zapoznanie się ze standardem IEEE 802.3af i wariantami jego realizacji	x	x		
Projekt układu detekcyjnego urządzenia PoE	x	x		
Symulacja działania i odpowiedni dobór elementów według klasy energetycznej		x		
Projekt przetwornicy step-down	x	x		
Symulacja działania przetwornicy i dobór elementów		x		
Symulacja finalnego układu		x		
Schemat elektryczny płytki PCB	x	x		
Layout płytki PCB	x			
Montaż elementów na wydrukowanej płytce	x	x		x
Testowanie działania poszczególnych sekcji płytki	x	x		x
Testowanie układu z wykorzystaniem przełącznika sieciowego obsługującego standard PoE	x	x		
Rekonesans odnośnie istniejących generycznych implementacji systemów operacyjnych czasu rzeczywistego typu open-source na mikrokontrolery			x	x
Rekonesans odnośnie istniejących aplikacji klienta LwM2M typu open-source				x
Uruchomienie przykładowego programu klienta LwM2M na płytce rozwojowej			x	x
Zintegrowanie przykładowego programu sensora ENS210 z płytką esp32			x	x
Zintegrowanie programu klienta LwM2M z obsługą sensora ENS210			x	x
Ugenerycznienie aplikacji w celu możliwości skompilowania programu na inne architektury procesorów			x	x
Dokumentacja	x	x	x	x

2. SPRZĘT

Wybór komponentów

- Wybór elementów elektronicznych wykonanych w THT jest spowodowany posiadaniem takich elementów przez Nas co znacznie ograniczyło koszty projektu.
- Układ detekcji PoE został zrealizowany za pomocą tranzystorów a nie gotowego układu, gdyż nie był on dostępny.
- Wybór przetwornicy BD9G341EFJ jest spowodowany brakiem dostępności na polskim rynku stabilizatora step-down o wymaganych w projekcie parametrach (nap. wej. 54 V), np. LM2576HVT-ADJ.
- Czujnik ENS210 został udostępniony przez prowadzącego laboratorium. Mniej oczywistym powodem wyboru tego sensora był fakt, iż ten sensor wyróżnia się wysoką liniowością w stosunku do innych rodzajów sensorów, a jego zakres pracy ($-40\text{ }^{\circ}\text{C}$ – $100\text{ }^{\circ}\text{C}$) jest wystarczający w stosunku do innych rodzajów sensorów (np. -600°C – $2000\text{ }^{\circ}\text{C}$ dla termopary).
- Z powodu racjonalnego kosztu projektu nie zamontowaliśmy na płytce układu ENC28J60x-SP oraz transformatorów. Nie są one kluczowe do poprawnego działania czujnika.
- Płytką rozwojową ESP32-DevkitC-V4 została wybrana jako płytka prezentująca działanie układu, ponieważ posiadaliśmy w zapasie wystarczającą liczbę tych układów w przypadku ich uszkodzenia podczas tworzenia projektu. Ostatecznie aplikację (po drobnych modyfikacjach plików konfiguracyjnych) można uruchomić na prawie dowolnym zestawie rozwojowym z łącznością bezprzewodową, o ile posiada on wsparcie systemu operacyjnego Zephyr OS.

Opis schematu

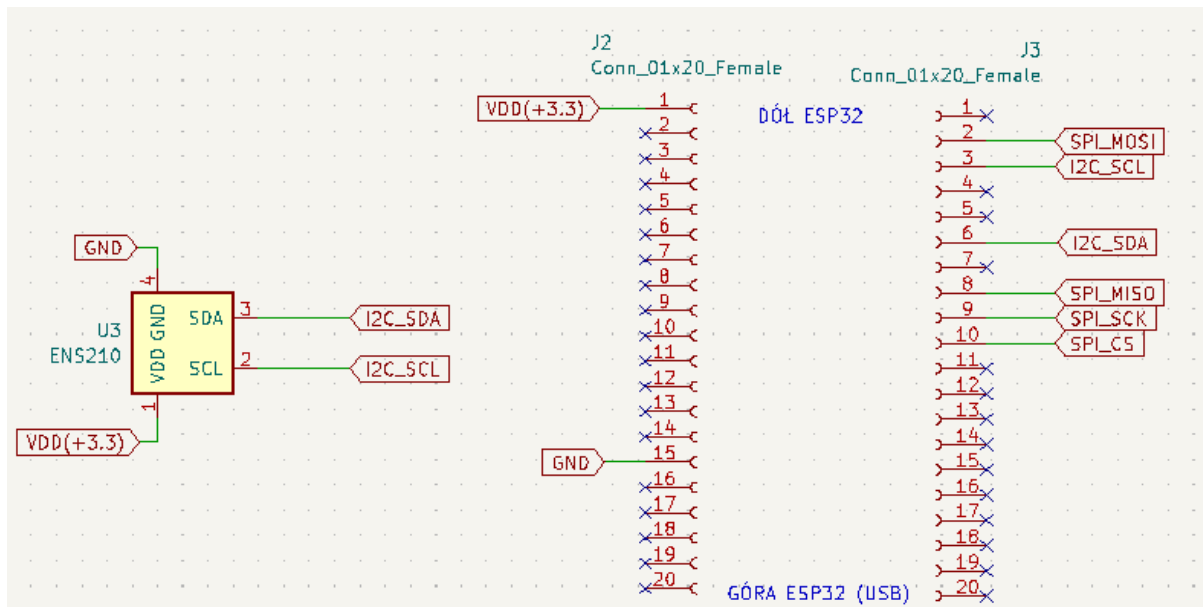
Obwód projektu podzielono na cztery segmenty:

1. zasilający płytkę – składa się on z dwóch mostków Graetza zapobiegających odwrotnej polaryzacji obwodu i przewidzianych pól lutowniczych na dwa transformatory separujące sygnał danych AC od zasilania DC przesyłanych tymi samymi parami przewodów (nie wykorzystane w projekcie). Dostarczenie zasilania do płytki poprzez gniazdo Ethernet umożliwia układ identyfikujący moduł jako urządzenia spełniającego standard PoE. Zapewnia on impedancje wejściową w granicach $25\text{ k}\Omega$ oraz odpowiedni pobór prądu w trakcie klasyfikacji urządzenia.
2. przetwarzający napięcie na odpowiednią wartość – układ przetwornicy obniżającej napięcie DC z zakresu przewidzianego w standardzie 44 V – 57 V na napięcie 3.3 V zasilające mikrokontroler ESP32 oraz wykorzystany sensor ENS210.
3. logiczny – znajduje się w nim mikrokontroler ESP32 oraz czujnik ENS210.
4. Ethernet PHY – przewidziane pola lutownicze na komunikację Ethernet poprzez SPI.

The schematic diagram illustrates the internal components and connections of the ENC28J60 module. At the center is the ENC28J60 microcontroller (U4). Key components and connections include:

- Power Supply:** A 3.3V VDD supply is connected to the VDD pin (pin 20) and the VDDPILL pin (pin 21). A 10μF capacitor (C5) is connected between VDD and GND.
- Crystal Oscillator:** A crystal (Y1) is connected to the OSC1 (pin 23) and OSC2 (pin 24) pins. A 18pF capacitor (C7) is connected between each oscillator pin and GND.
- Reset:** A 10K resistor (R18) is connected between the RESET pin (pin 10) and the 3.3V supply.
- LED:** The LEDA (pin 27) and LEDB (pin 26) pins are connected to the 3.3V supply.
- Host Interface:**
 - TPIN+ and TPIN-:** Connected to pins 13 and 12 respectively.
 - PHY_TX+ and PHY_TX-:** Connected to pins 19 and 18 respectively.
 - PHY_RX+ and PHY_RX-:** Connected to pins 17 and 16 respectively.
 - PHY_TCT_TX and PHY_TCT_RX:** Connected to pins 15 and 14 respectively.
- Other Pins:**
 - SPL_MOSI (pin 7), SPL_MISO (pin 8), SPL_SCK (pin 9), and SPL_CS (pin 4):** These pins are connected to the host system's SPI interface.
 - INT (pin 4) and WOL (pin 5):** These pins are connected to the host system's interrupt and wake-on-lan pins.
 - VDDOSC (pin 25) and VSSOSC (pin 22):** These pins are connected to the host system's oscillator supply and ground.
 - CLKOUT (pin 3):** This pin is connected to the host system's clock output.
 - VSS (pin 2) and VSSPILL (pin 21):** These pins are connected to the host system's ground.
 - RBIAS (pin 14) and VCAP (pin 1):** These pins are connected to the host system's bias and capacitor pins.

Rysunek 3. Schemat połączeń Ethernet PHY



Rysunek 4. Schemat połączeń sensora oraz mikrokontroler

Symulacje

W celu sprawdzenia oraz doboru elementów wykonane zostały symulacje poszczególnych układów oraz całej płytki w programie LTspice.

Sprawdzona została poprawności działania układu identyfikującego oraz klasyfikującego urządzenie jako urządzenie PD (Powered Device) standardu PoE poprzez odwzorowanie pomiaru impedancji wejściowej przez PSE (Power Sourcing Equipment – switch PoE). Pomiar ten polega na zmierzeniu poboru prądu dla dwóch ustalonych napięć i na podstawie ich różnicy wyznaczenia impedancji PD.

Wyniki pomiaru impedancji:

$$I_{PSE} = 120 \text{ uA dla } V_{PSE} = 4 \text{ V}$$

$$I_{PSE} = 278 \text{ uA dla } V_{PSE} = 8 \text{ V}$$

$$\frac{\Delta V_{PSE}}{\Delta I_{PSE}} = 25,3 \text{ k}\Omega$$

Wartość ta spełnia wymagania, gdyż mieści się w przedziale 19 k Ω – 26.5 k Ω .

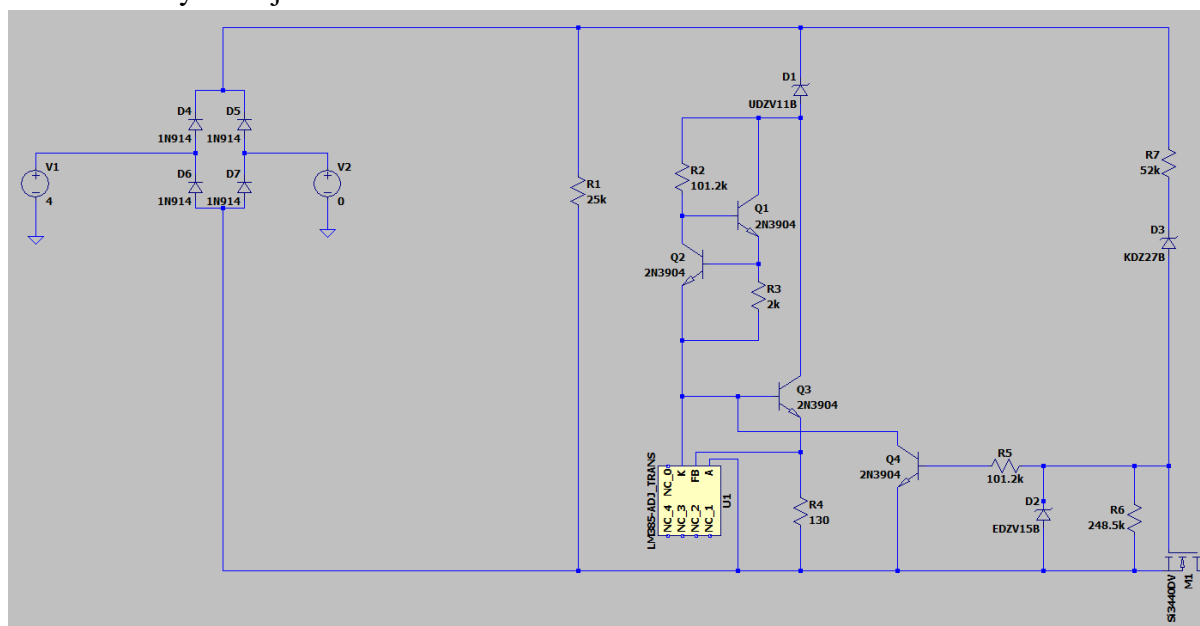
Następnie zmierzony został pobór prądu dla napięcia 20 V pozwalający na odpowiednią klasyfikację mocową urządzenia zgodnie z tabelą 2. Pobór prądu wyniósł 10.6 mA co oznacza klasę 1 i dostępną z PSE moc maksymalnie do 3.84 W.

Tabela 2. Klasyfikacja urządzenia PoE względem pobieranego prądu

Measured classification current	Classification
0mA to 5mA	Class 0
>5mA and <8mA	Class 0 or 1
8mA to 13mA	Class 1
>13mA and <16mA	Class 0, 1 or 2
16mA to 21mA	Class 2
>21mA and <25mA	Class 0, 2 or 3
25mA to 31mA	Class 3
>31mA and <35mA	Class 0, 2 or 3
35mA to 45mA	Class 4
>45mA and <51mA	Class 0 or 4
$\geq 51\text{mA}$	Class 0

Class	Purpose	Power at the PD input
0	15.4 W	0.44 to 12.95 W
1	4.0 W	0.44 to 3.84 W
2	7.0 W	3.84 to 6.49 W
3	15.4 W	6.49 to 12.95 W
4	Like class 0	Reserved for future applications

Schemat do symulacji:

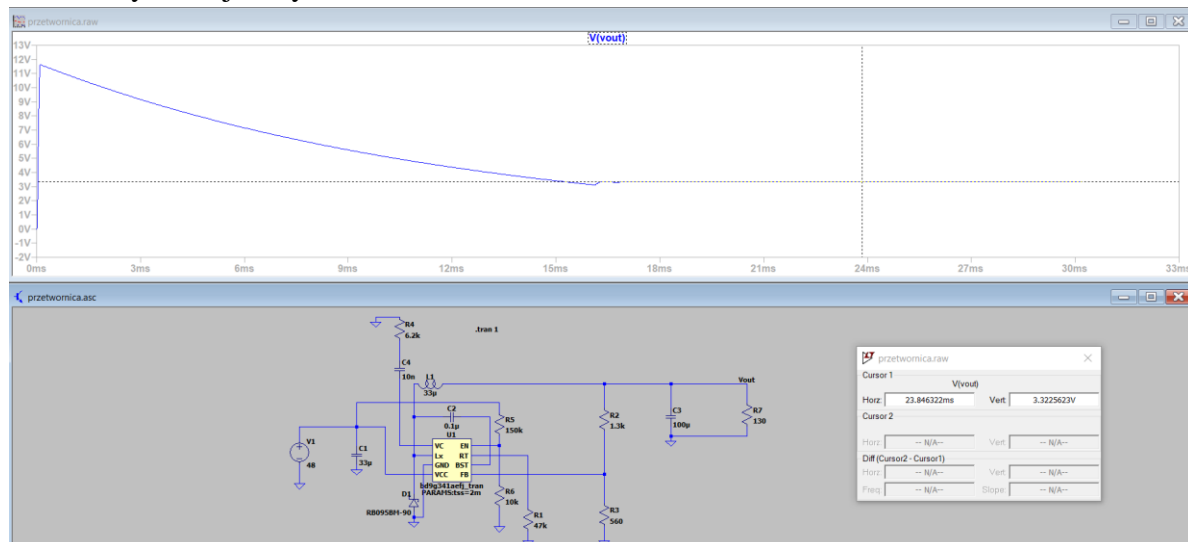


Rysunek 5. Schemat symulacji obwodu detekcji PoE

Wykonana została również symulacja stabilizatora napięcia, w celu dobrania elementów i weryfikacji napięcia wyjściowego 3.3 V DC dla napięcia wejściowego 48 V DC oraz obciążenia w postaci rezystora 130 Ω .

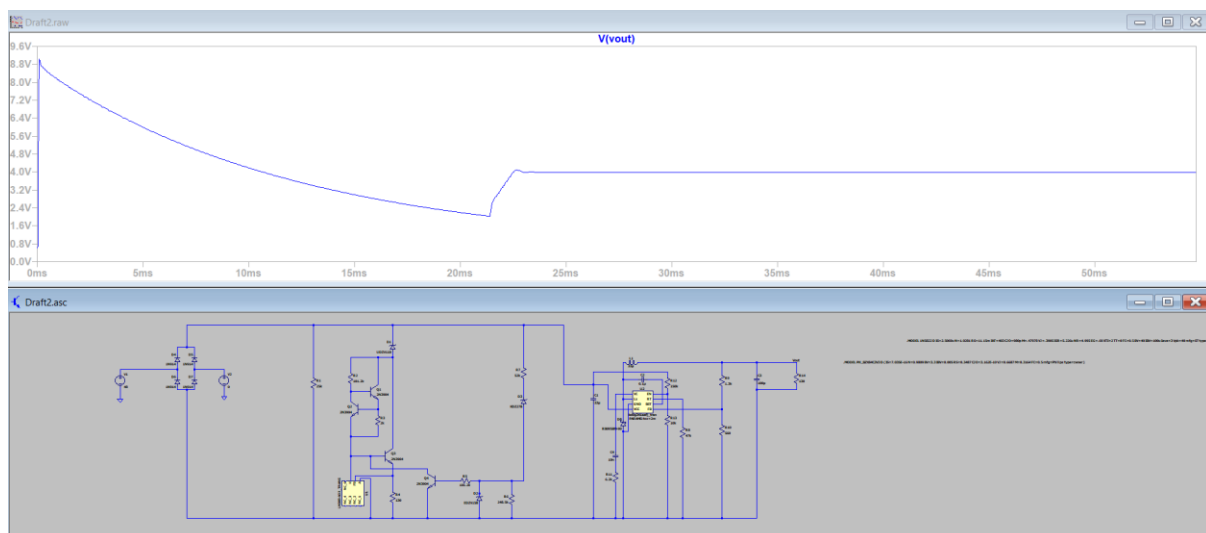
Na wyjściu można zaobserwować stałe napięcie 3.3 V.

Schemat symulacji i wyniki:



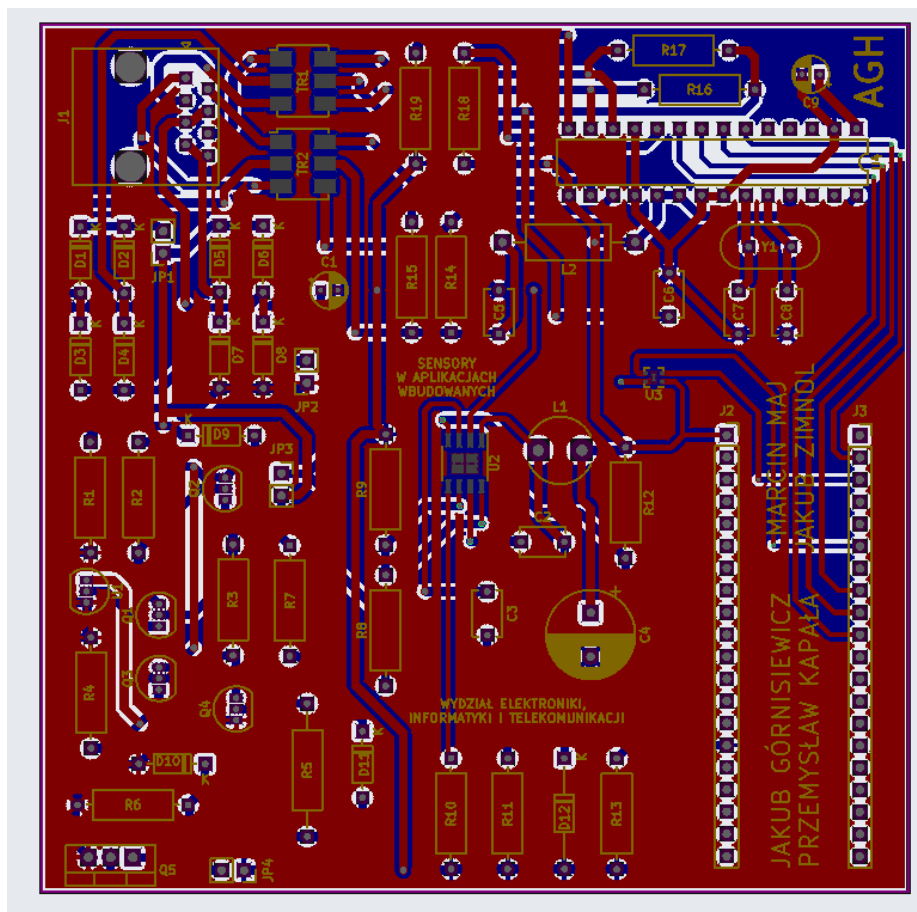
Rysunek 6. Symulacja przetwornicy obniżającej napięcie

Ostatnia symulacja została przeprowadzona dla połączonego układu detekcyjnego oraz obniżającego napięcie. Tranzystor MOSFET (Q5 ze schematu na rys. 2) został zastąpiony zwarciem.



Rysunek 7. Symulacja połączonych sekcji

Layout



Rysunek 8. Layout płytki

3. OPROGRAMOWANIE

Już od samego początku projekt był planowany z myślą o uniwersalności i łatwym portowaniu na inne architektury sprzętowe - z tego też powodu zdecydowaliśmy się na wykorzystanie systemu operacyjnego Zephyr OS. Na nasze szczęście Zephyr posiada specjalne API do obsługi sensorów i odczytywania danych oraz, przede wszystkim, wbudowaną obsługę posiadanego przez nas sensora ENS210. Ponieważ temperatura czy wilgotność w pomieszczeniu serwerowym nie powinny szybko i drastycznie zmieniać swoich wartości, zdecydowaliśmy o niefiltrowaniu otrzymywanych danych z sensora w urządzeniu, ponieważ w razie potrzeby taką operację można wykonać już po stronie serwera. Aby okresowo gromadzić dane oraz przysyłać je do serwera IoT, zdecydowaliśmy się wykorzystać aplikację klienta LwM2M (*Lightweight Machine-to-Machine*) Anjay, a właściwie jego implementację o nazwie Anjay-zephyr-client. LwM2M to protokół komunikacyjny oparty na architekturze klient-serwer, który umożliwia zdalne zarządzanie urządzeniami IoT poprzez monitorowanie, konfigurację i aktualizację ich zasobów (np. Firmware Update Over-the-Air). Jest to idealny protokół dla aplikacji opartych o telemetrię, czyli zdalne pobieranie i analizowanie danych z sensorów.

Sposób odczytywania danych z sensora, przechowywania ich w urządzeniu i przysyłania do serwera jest ściśle związany z protokołem LwM2M. Odczyt danych z sensora

odbywa się co sekundę (wartość konfigurowalna) i wartość ostatniego pomiaru jest zapisywana w pamięci urządzenia w Resource 5700 odpowiedniego Obiektu (/3303 w przypadku temperatury oraz /3304 w przypadku wilgotności). Aby urządzenie wysłało ostatnią zmierzoną wartość do serwera, z poziomu interfejsu graficznego Coiote DM należy wykonać operację Read, która informuje urządzenie, aby jednorazowo wysłał aktualną wartość konkretnego Resource lub całego Obiektu, albo operację Observe, która informuje urządzenie, aby okresowo wysyłało dane do serwera. Operacja Send nie została zaimplementowana, ponieważ uważaliśmy, iż urządzenie w pomieszczeniu serwerowym powinno mieć stabilne połączenie z Internetem, więc założyliśmy, że okresowe wysyłanie danych do serwera na prośbę serwera jest w pełni wystarczające. Więcej na temat protokołu LwM2M oraz biblioteki Anjay można przeczytać w [oficjalnej dokumentacji biblioteki](#).

Aktualnie projekt jest przystosowany do uruchomienia tylko na zestawie rozwojowym z serii ESP32. Aby uruchomić projekt na innych zestawach rozwojowych (posiadających łączność z Internetem), należy dodać odpowiednie pliki konfiguracyjne (pliki *.conf oraz *.overlay) do folderu Anjay-zephyr-client/agh_swaw_project/boards. Część z plików konfiguracyjnych jest wstępnie zaimplementowana w przykładach aplikacji Anjay (tylko część związana z warstwą sieciową), natomiast trzeba indywidualnie zintegrować wykorzystywany sensor z *devicetree* urządzenia.

Kod źródłowy oraz krótka instrukcja dot. budowania aplikacji i flashowania urządzenia znajduje się w [repozytorium GitHub](#).

ID	RESOURCE NAME	VALUE	LAST UPDATED	WIDGET	OPERATIONS
/5601	Min Measured ...	23.740625	2023-06-08, 12:53:22	[Widget Icon]	[Refresh] [Eye]
/5602	Max Measured...	23.75625	2023-06-08, 12:53:22	[Widget Icon]	[Refresh] [Eye]
/5605	Reset Min and ...	not applicable	-	[Widget Icon]	[Settings]
/5700	Sensor Value	26.209375	2023-06-08, 13:06:41	[Widget Icon]	[Refresh] [Eye] [Read resource]
/5701	Sensor Units	Cel	2023-06-08, 12:53:22	[Widget Icon]	[Refresh] [Eye]

ID	RESOURCE NAME	VALUE	LAST UPDATED	WIDGET	OPERATIONS
/5601	Min Measured ...	61.550748	2023-06-08, 12:53:22	[Widget Icon]	[Refresh] [Eye]
/5602	Max Measured...	62.074152	2023-06-08, 12:53:22	[Widget Icon]	[Refresh] [Eye]
/5605	Reset Min and ...	not applicable	-	[Widget Icon]	[Settings]
/5700	Sensor Value	52.195878	2023-06-08, 13:06:41	[Widget Icon]	[Refresh] [Eye] [Read resource]
/5701	Sensor Units	%RH	2023-06-08, 12:53:22	[Widget Icon]	[Refresh] [Eye]

Rysunek 9. Widok obiektów temperatury i wilgotności z poziomu serwera Coiote DM oraz operacji Read resource

4. URUCHOMIENIE

Płytkę udało się uruchomić oraz działa ona poprawnie, zgodnie z założeniami. Współpracuje ona z częścią programową a odczyt danych widziany jest na serwerze.

Problemy i komplikacje

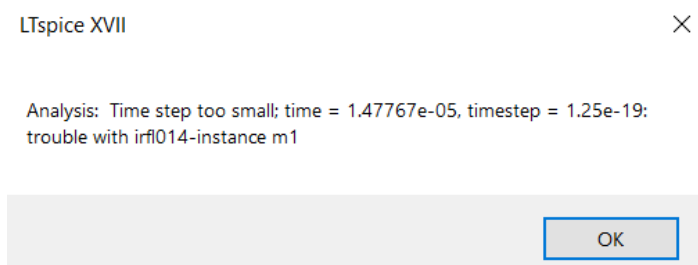
Część sprzętowa

Pierwszym poważnym problem na jaki trafiliśmy podczas projektowania obwodu elektrycznego był zaszyfrowany model LTSpice wykorzystanej przetwornicy. Z tego powodu konieczne było poproszenie producenta o udostępnienie modelu. Niestety informacje zwrotną otrzymaliśmy dopiero po dwóch tygodniach przez co płytkę musieliśmy wysłać do produkcji bez ostatecznej weryfikacji poprawności działania tego segmentu.



Rysunek 10. Informacja zwrotna od producenta

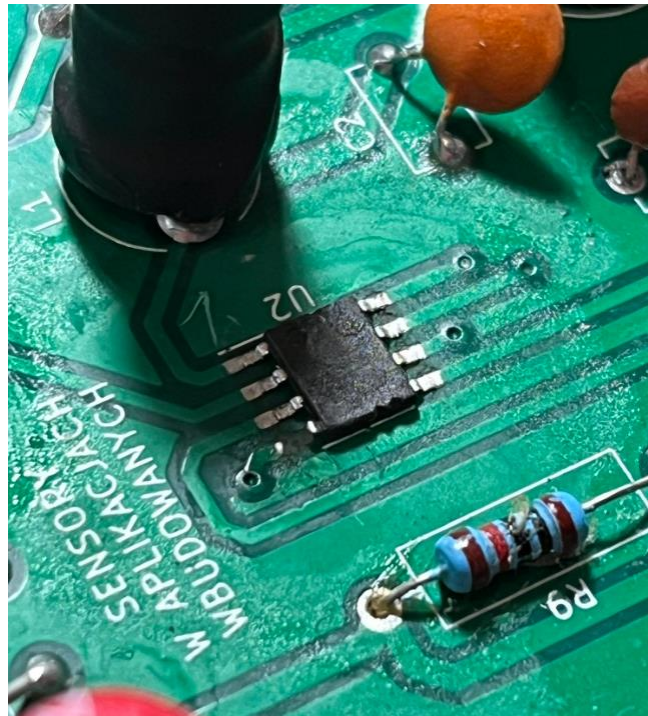
W trakcie symulacji ostatecznej wersji układu napotkaliśmy również na problem techniczny z samym symulatorem LTSpice, gdyż nie był on w stanie wykonać obliczeń dla całego układu. Z tego powodu tranzystor został zastąpiony zwarcie, ponieważ mimo zmiany jego modelu błąd dalej występował.



Rysunek 11. Błąd LTSpice

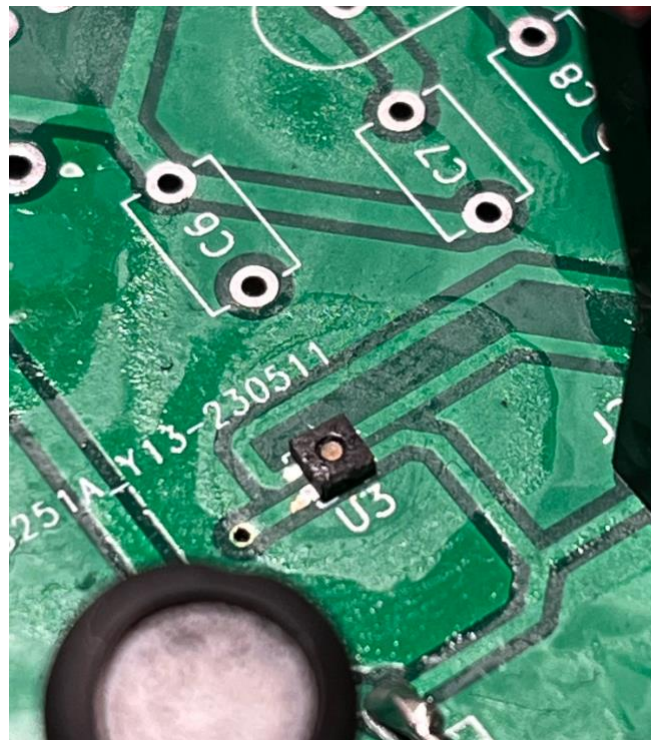
Kolejnym ciekawym (i kosztownym) doświadczeniem było popełnienie błędu polaryzacji napięcia podczas weryfikacji działania płytki przez co uszkodzony został stabilizator oraz efektownie wybuchł kondensator. Był to nasz pierwszy wybuch, dzięki czemu dowiedzieliśmy się, że kondensator faktycznie ma dwie okładziny a napięcie maksymalne ma znaczenie 😊.

Dodatkowo podczas wylutowywania stabilizatora urwał się pad przez co konieczne było przelutowywanie wszystkich komponentów występujących w układzie na nową płytkę.



Rysunek 12. Uszkodzony printfoot stabilizatora

Kamieniem milowym był również montaż sensora do płytki PCB. Po próbach przylutowania go za pomocą tradycyjnej lutownicy oporowej zdecydowaliśmy użyć do tego „Hot-air’a”. Była to Nasza pierwsza styczność z takim urządzeniem.



Rysunek 13. Przylutowany sensor

Podczas weryfikacji również natknęliśmy się na problem rezystancji wewnętrznej układu, dotychczasowo wiedzieliśmy o niej tylko „teoretycznie”. W tym przypadku sama sekcja zasilająca PoE „widziała” 25 k Ω i działała poprawnie dla odłączonej reszty układu. Weryfikacja poprawności działania samej sekcji obniżającej napięcie również przebiegła pomyślnie. Natomiast gdy obie sekcje zostały połączone, pojawiał się problem i płytka przestawała działać.

Dzięki pomocy prowadzącego laboratorium odkryto, iż powodem tej anomalii może być wpływający na rezystancję wejściową dzielnik rezystancyjny podłączony do wejścia „EN” stabilizatora. Transzystor MOSFET, który w fazie detekcji miał odłączać sekcje obniżającą napięcie był zwarty bez napięcia polaryzującego. Rozwiązaniem tej sytuacji była zmiana tranzystora w układzie PoE z IRF9530N (kanał typu P) na RFP10N15 (kanał typu N).

Sam layout płytki mógłby zostać lepiej zrealizowany. Był to nasz pierwszy projekt PCB, który miał zostać zrealizowany przez fabrykę. Popelniliśmy kilka błędów przy projektowaniu:

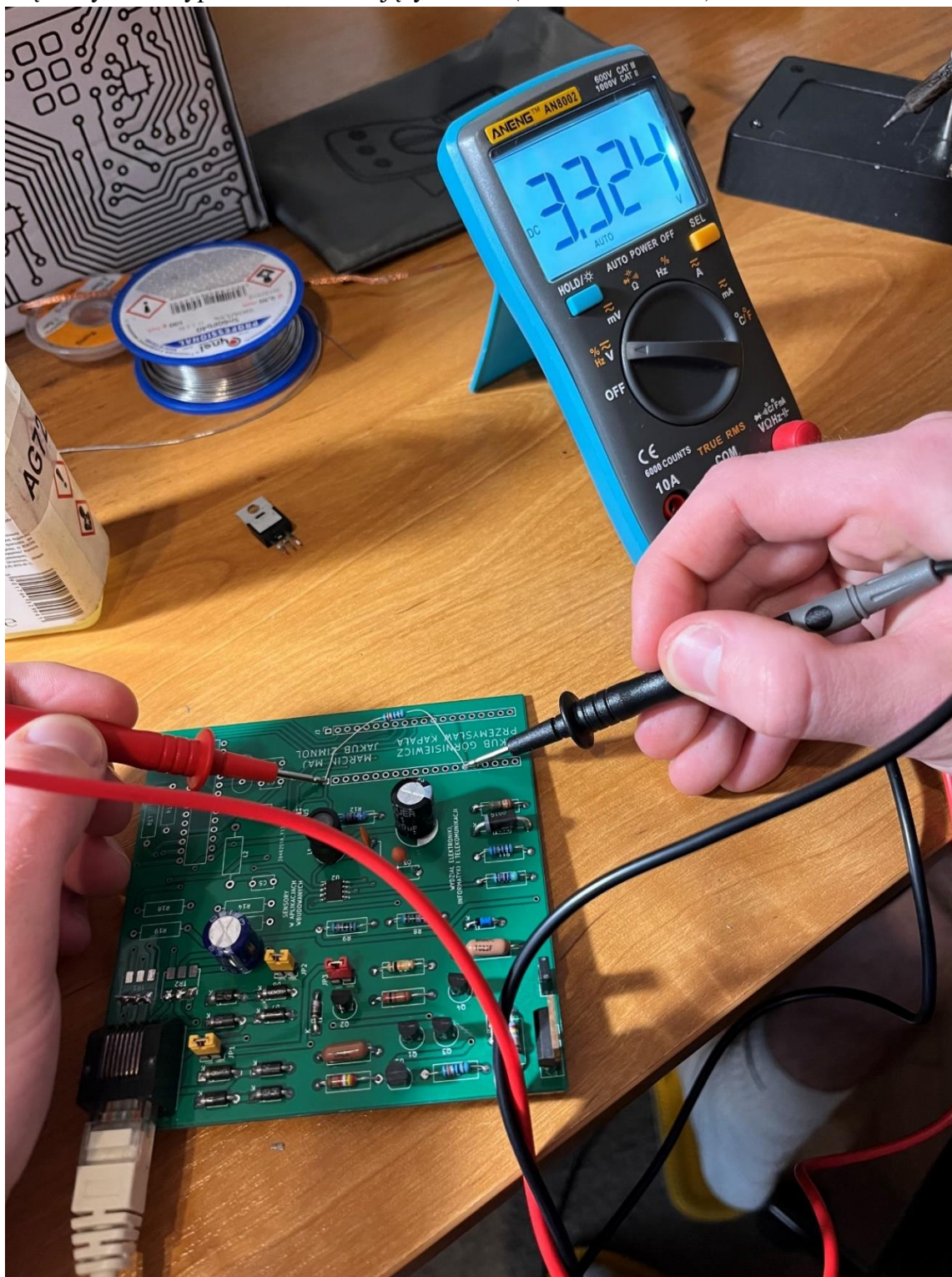
- przyzwyczajenie do wytwarzania płytek metodą termotransferu podświadomie staraliśmy się zbędnie umieszczać wyprowadzenia elementów na dolnej warstwie,
- brak otworów montażowych płytki przez co nie można przykręcić nóżek do płytki
- masa wysokiego i niskiego napięcia zostały rozlane na dwóch stronach płytki, lepszym rozwiązaniem byłoby rozlanie odpowiedniej masy na obszarze, gdzie tylko ona występuje
- z powodu szybkiego czasu realizacji plików gerber, poprowadzenie ścieżek nie jest idealne, można by było lepiej wydzielić sekcje
- zły rozstaw nóżek ESP, przez co sam moduł jest wpięty za pomocą przewodów

Część oprogramowania

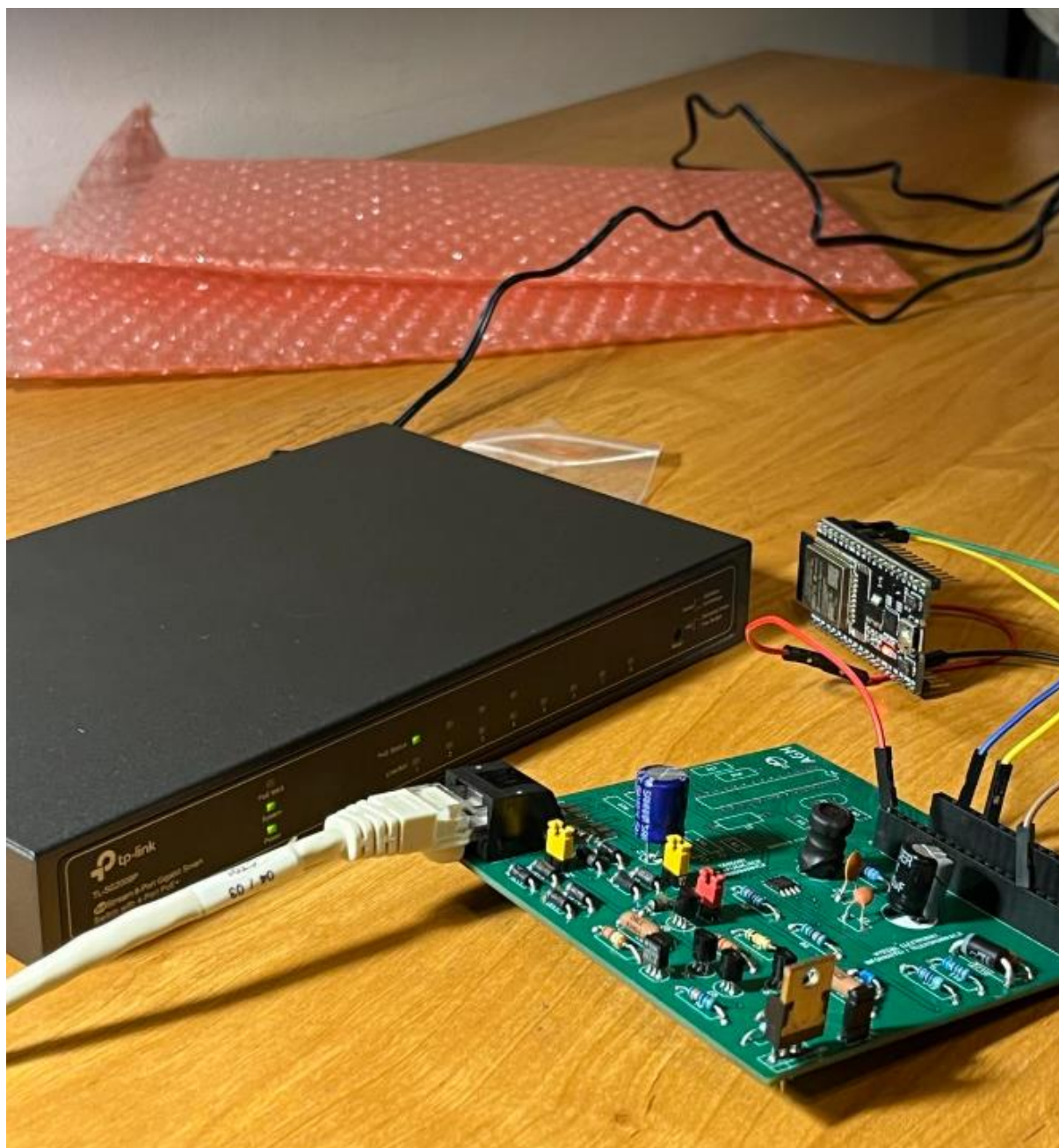
Implementacja części programowej w dużej mierze odbyła się bez problemów, ponieważ już wcześniej mieliśmy (prywatnie) doczynienia z systemem operacyjnym Zephyr OS czy protokołem LwM2M. Jednakże podczas uruchamiania przykładowych aplikacji Zephyr OS na posiadanej płytce rozwojowej z serii ESP32 zauważyliśmy, iż bezpośrednio po reboocie urządzenia płytka nie zawsze od razu łączy się do Internetu, pomimo że z sukcesem łączy się do wskazanej przez nas sieci Wi-Fi. Rozwiązaniem tego problemu okazało się resetowanie interfejsu sieciowego urządzenia kilka sekund po włączeniu jego zasilania, jednakże ta opcja - dostępna z poziomu interfejsu szeregowego – nie była dla nas dostępna w przypadku, gdy urządzenie było zasilane z zewnętrznego źródła (PoE), a nie przewodu USB. Ponieważ nie mamy pewności, czy problem ten dotyczy wszystkich płytek rozwojowych z serii ESP32 czy tylko posiadanych przez nas egzemplarzy, tymczasowo do kodu programu wstawiliśmy instrukcje, które kilka sekund po reboocie urządzenia resetują interfejs sieciowy, co powoduje połączenie się urządzenia do sieci. Problemem prawdopodobnie mogą być np. *race conditions* występujące pomiędzy sterownikami do Wi-Fi oraz warstwą sieciową biblioteki Anjay.

Weryfikacja

Weryfikacja działania płytki polegała na zmierzeniu napięcia na rezystorze 130 Ω podłączonym do wyprowadzeń zasilających ESP (+3.3V oraz GND).



Rysunek 14. Weryfikacja napięcia



Rysunek 15. Finalny wygląd urządzenia. Kontrolka (dioda czerwona) ESP świeci sygnalizując działanie, kontrolka PoE na Switchu również sygnalizuje prace (środkowa)



Bibliografia

„White Paper Power over Ethernet IEEE 802.3af.” Hirschmann a Belden Brand, 2011,
www.belden.com/hubfs/emea/resources/Picture%20Park%20Assets/Files%20for%20Redirection/WP_%20Power%20over%20Ethernet%20%28PoE%29%20IEEE%20802.3af_Original_15240.pdf