

POLITECHNIKA OPOLSKA
WYDZIAŁ ELEKTROTECHNIKI AUTOMATYKI I
INFORMATYKI



INFORMATYKA
KOMPUTEROWE WSPOMAGANIE PROJEKTOWANIA

PRACA MAGISTERSKA

KOMUNIKACJA SIECIOWA W TECHNOLOGII
1-WIRE

Promotor: dr inż. Sławomir Zator

Autor: Maciej Marek Róg

Opole 2008

Spis treści

1.	Wstęp.....	4
2.	Cel i zakres pracy	5
3.	Przegląd literatury	6
3.1.	Przegląd metod komunikacji stosowanych w inteligentnych budynkach.....	6
3.2.	Protokoły komunikacyjne stosowane w instalacjach BAS/BMS	9
3.2.1.	Wprowadzenie	9
3.2.2.	Protokół BACnet	10
3.2.3.	Protokół LonTalk	11
3.2.4.	Protokół PROFIBUS	12
3.2.5.	Protokół EIB	14
3.2.6.	Protokół MODBUS	16
3.2.7.	Protokół CAN	17
3.2.8.	Protokół CANopen	19
3.2.9.	Protokół X10	22
4.	Technologia 1-Wire.....	24
4.1.	Wprowadzenie	24
4.2.	Opis działania układów 1-Wire	25
4.2.1.	Zasilanie	25
4.2.2.	Sygnały czasowe	27
4.2.3.	Unikalny identyfikator	31
4.2.4.	Obsługa błędów	32
4.2.5.	Korekta pomiarów temperatury.....	34
4.3.	Opis protokołu 1-Wire	37
4.3.1.	Warstwy protokołu	37
4.3.2.	Medium	39
4.3.3.	Prędkość działania	40

4.3.4.	Topologia sieci 1-Wire	41
4.3.5.	Zasięg sieci	43
5.	Układy pracujące w technologii 1-Wire.....	46
5.1.	Wprowadzenie	46
5.2.	Opis układów 1-Wire	47
5.2.1.	Układy pamięciowe	47
5.2.2.	Czujniki temperatury	48
5.2.3.	Interfejsy Komunikacyjne	50
5.2.4.	Przetworniki A/C.....	51
5.2.5.	Układy przetwarzania czasu	52
5.2.6.	Przełączniki	53
5.2.7.	Pomiar parametrów mikroklimatu z wykorzystaniem układów 1-Wire	54
6.	Obsługa urządzeń 1-Wire.....	56
6.1.	Wprowadzenie	56
6.2.	Dostępne oprogramowanie realizujące komunikację siecią MicroLAN ...	57
6.3.	Opis środowiska LabVIEW	68
6.4.	Obsługa układów 1-Wire w środowisku LabVIEW	74
6.5.	Opis aplikacji	75
6.6.	Pomiary i kalibracja	86
7.	Podsumowanie i wnioski.....	89
8.	Spis rysunków	92
9.	Spis tabel	94
10.	Bibliografia.....	95
	Załącznik 1	98

1. Wstęp

W obecnym czasie można zauważyć bardzo dynamiczny rozwój technologiczny, w szczególności automatyki. Rozwiązania takie znaleźć można już praktycznie we wszystkich dziedzinach życia. Dotychczas sterowanie czy automatyka były domeną fabryk, zakładów przemysłowych, w których wykorzystywano je do zarządzania procesami produkcji. Obecnie coraz częściej instalacje takie stosuje się w budynkach użyteczności publicznej, czy w gospodarstwach domowych. Dzięki zastosowaniu automatyki zyskuje się, oprócz komfortu, oszczędność energii, co ma bezpośredni wpływ na stan środowiska naturalnego.

Inteligentny system sterowania, aby działał skutecznie, musi reagować w odpowiednim czasie na dostarczane mu dane. Niniejsza praca zajmuje się wycinkiem tego zagadnienia, jakim jest akwizycja danych, a dokładniej pomiar temperatury. Do tego celu wykorzystano cyfrowe czujniki 1-Wire serii DS18B20. Zbudowany układ pomiarowy, w skład którego wchodzi: czujniki temperatury, kontroler magistrali 1-Wire i komputer PC, współpracuje z napisaną w środowisku LabVIEW aplikacją.

W rozdziale 3 omówiono metody komunikacji stosowane w instalacjach inteligentnych budynków. W szczególności nacisk położono na przedstawienie charakterystyki protokołów transmisji. W rozdziale 4 opisano technologię 1-Wire: sposób jej działania i możliwości. Zobrazowano również w sposób szczegółowy sam protokół: jego warstwy, topologię połączeń i prędkość transmisji.

W rozdziale 5 zaprezentowano układy przystosowane do pracy w technologii 1-Wire. Opis przedstawia zestawienie parametrów wersji poszczególnych urządzeń producenta układów, firmy Dallas-Maxim. Ponadto zaproponowano przykład układów pomiaru wilgotności i stężenia CO₂ innych producentów, które można wykorzystać w sieci 1-Wire z pomocą przetwornika A/C.

Rozdział 6 zawiera opis działania aplikacji, która służy do współpracy z czujnikami DS18B20. Program umożliwia wykonywanie pomiarów, kalibrację i korektę pomiarów. W rozdziale zaprezentowano środowisko LabVIEW, w którym aplikacja została napisana. Rozdział zawiera prezentacje wykonanych pomiarów kalibracyjnych i uzyskanych dzięki nim poprawionych wyników.

2. Cel i zakres pracy

Celem pracy jest zbadanie możliwości wykorzystania technologii 1-Wire do budowy sieci czujników mierzących parametry mikroklimatu pomieszczenia. Wynikiem pracy jest projekt układu, realizującego pomiar i obróbkę wybranych wielkości fizycznych – temperatury.

Zakres pracy obejmuje:

- przegląd literatury,
- projekt sieci czujników,
- opis stanowiska pomiarowego,
- opracowanie aplikacji,
 - zbieranie pomiarów,
 - przetwarzanie wyników,
 - prezentacja danych,
- wyniki testowania i analizę działania programu.

Do analizy wykorzystania technologii 1-Wire użyto graficznego środowiska programistycznego LabVIEW. Oprogramowanie wyposażone jest w narzędzia wspomagające gromadzenie, przetwarzanie i analizę danych pomiarowych. Narzędzia te pozwalają programiście na zautomatyzowanie wielu czynności projektowych. Graficzne tworzenie aplikacji umożliwia łatwe dostosowywanie aplikacji do wprowadzanych zmian, przez co nie ma potrzeby analizy setek linii kodu. Biorąc pod uwagę charakterystykę i elastyczność urządzeń pracujących w technologii 1-Wire, jest to niewątpliwą zaletą.

3. Przegląd literatury

3.1. Przegląd metod komunikacji stosowanych w inteligentnych budynkach

Pomiar temperatury pomieszczenia jest jednym z elementów instalacji zarządzania ogrzewaniem w budynkach. Systemy automatyki budynku, określane w języku angielskim skrótem BAS (ang. *Building Automation System*), jest wielopoziomowym rozwiązaniem służącym do monitorowania pomieszczeń. Obejmuje ono poniższe grupy [21]:

- komfort cieplny (ogrzewanie, wentylacja, klimatyzacja),
- energia (oświetlenie, windy),
- nadzór (autoryzacja, kontrola dostępu),
- bezpieczeństwo (oddymianie, inst. przeciw pożarowe).

Początkowo każdy system istniał w budynku jako niezależna instalacja. Różnorodność możliwości oferowanych przez poszczególne rozwiązania, dostarczone najczęściej przez różnych producentów, powodowała niepotrzebne podnoszenie kosztów (sprzęt i oprogramowanie) a także brak możliwości sprawowania zcentralizowanej kontroli nad systemem jako całością. Z upływem lat powstało rozwiązania integrujące wszystkie podsystemy w jedną całość, BMS (ang. *Building Management System*). System ten można podzielić na dwie niezależne części [1]:

- centralne sterowanie i nadzór instalacji technicznych - BAS,
 - wytwarzanie ciepła i chłodu, wentylacja i klimatyzacja,
 - oddymianie, kontrola instalacji elektrycznej,
- monitoring instalacji bezpieczeństwa XMS¹,
 - kontrola dostępu, identyfikacja (CCTV)²,
 - sygnalizacja pożaru.

Każdy element systemu BMS pracuje w sposób autonomiczny, komunikuje się pomiędzy poszczególnymi podsystemami w celu wymiany kluczowych informacji. Nadzór nad działaniem systemu prowadzony jest z jednego miejsca. Integracja wszystkich technologii i wysoka elastyczność obniża koszty eksploatacji, zapewnia sprawne działanie i zapewnia odpowiedni poziom bezpieczeństwa użytkownikom systemu. Otwarcie systemu, jest kolejnym krokiem w „ewolucji” działania systemów monitoringu budynków. W szczególności otwarte protokoły komunikacyjne umożliwiają łączenie w całość

¹ ang. XMS – Excel Security Manager

² ang. CCTV - Closed-circuit television

podzespołów dostarczonych od różnych producentów. Dzięki temu użytkownik może wybrać odpowiednio najlepsze dla siebie rozwiązania w ramach budżetu, którym dysponuje. Protokoły komunikacyjne w budynkach z wdrożonym systemem BMS, oparte są najczęściej o okablowanie strukturalne.

System połączeń, zwany topologią sieci, determinuje zastosowanie wybranego protokołu komunikacyjnego. Wyróżniamy cztery rodzaje budowy sieci: drzewo, magistrała, pierścień i gwiazda. Dobór odpowiedniej topologii związany jest stricte z wybranym przez użytkownika protokołem komunikacyjnym. Szeroko stosowane obecnie okablowanie strukturalne, pozwala na przesył zarówno danych informatycznych, jak i sygnałów powiązanych z instalacją nadzoru budynku. Zastosowanie takiego rozwiązania zapewnia dużą elastyczność, umożliwia montaż i demontaż elementów sterujących w dowolnym miejscu w sieci. Dodatkowym atutem zastosowania jednego rodzaju okablowania, jest znaczna redukcja kosztów. Okablowanie strukturalne składa się z:

- kabla sieciowego, tworzącego szkielet sieci,
- punktów dostępowych i rozdzielczych,
- instalacji okablowania poziomego.

Do budowy sieci w budynkach stosuje się najczęściej nieekranowany przewód, o czterech parach przewodów, zwany popularnie skrętką. Ten rodzaj kabla wykorzystywany jest głównie do okablowania poziomego. Szkielet sieci można zbudować przy wykorzystaniu skrętki, coraz częściej jednak w dużych instalacjach stosuje się światłowody. Inwestycja w szkielet zbudowany w oparciu o światłowód jest rozwiązaniem zapewniającym komfort zapasu przepustowości na kilka-kilkanaście najbliższych lat. Dodatkową zaletą kabli światłowodowych jest ich odporność na zakłócenia elektromagnetyczne i znacznie większy zasięg działania, przy przepustowości przewyższającej możliwości kabli miedzianych.

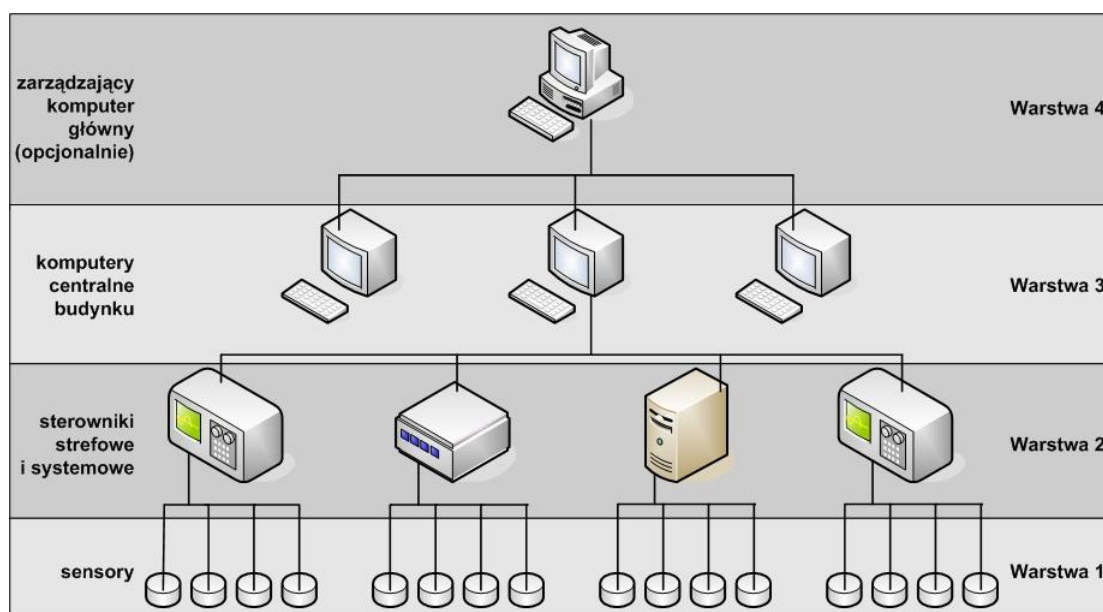
Działanie poszczególnych urządzeń, połączenia kablowe i protokoły używane podczas transmisji, są opisane przez model OSI³. Składa się on z siedmiu warstw [20]:

- aplikacji,
- prezentacji,
- sesji,
- transportowa,
- sieciowa,
- łącza danych,
- fizyczna.

³ ang. OSI - Open Systems Interconnection Basic Reference Model.

W inteligentnych budynkach liczba warstw jest zredukowana do pięciu. Do każdej warstwy przypisany jest pewien zestaw instrukcji i metod, za pomocą których możliwe jest operowanie na danych aktualnie znajdujących się w danej warstwie. Dane te są przekazywane bezpośrednio z warstwy niższej do warstwy wyższej (i na odwrót), dodatkowo wyposażone w pewną preambułę, która jest charakterystyczna dla danej warstwy. Takie rozwiązanie pozwala adoptować ten model do użycia z różnym rodzajem sprzętu i oprogramowania.

Łatwość implementacji modelu ISO/OSI pozwoliła na jego wykorzystanie w systemach BAS. W takim przypadku stosuje się uproszczoną wersję, trój lub czterowarstwową (rys. 3.1).



Rys. 3.1 Implementacja modelu ISO/OSI w systemach BAS [21]

Najniższa w hierarchii warstwa 1, zajmuje się odczytem analogowym bądź cyfrowym określonych parametrów mierzonych w budynku. Dane te w postaci sygnałów elektrycznych są przekazywane do warstwy 2. W przeciwieństwie do warstwy 1, urządzenia warstwy 2 mogą się ze sobą komunikować. Zadaniem urządzeń zlokalizowanych w tej warstwie jest sterowanie automatyką budynku, na podstawie odczytów przekazanych od czujników. W nowoczesnych systemach BAS stosuje się sterowniki DDC⁴, które pozwalają sterować lokalnie przydzielonymi do nich urządzeniami, z pominięciem komputerów warstwy 3. Warstwa 3 sprawuje kontrolę nad funkcjonowaniem budynku, na podstawie danych przekazanych z warstwy 2. Komputery, wyposażone w bezpieczną ilość pamięci nieulotnej, umożliwiają zbieranie i archiwizowanie danych o działaniu systemu i tych pochodzących z komunikacji między sterownikami. Dzięki temu, może skuteczniej sterować automatyką

⁴ ang. DDC – Direct Digital Control (bezpośrednie sterowanie cyfrowe)

budynku. Warstwa 4, stosowana opcjonalnie, obejmuje komputer nadzorujący prace na „zarządcami” z warstwy 3, kiedy w budynku znajduje się kilka takich komputerów.

3.2. Protokoły komunikacyjne stosowane w instalacjach BAS/BMS

3.2.1. Wprowadzenie

Do budowy systemów BAS/BMS, zwanych popularnie „inteligentnymi budynkami” stosuje się sprzęt jak i oprogramowanie. Te dwa pojęcia w dzisiejszym z informatyzowanym świecie nie mogą bez siebie istnieć i powinno się je traktować razem jako całość. W szczególności w przypadku inteligentnych budynków, mamy do czynienia z różnym rodzajem sprzętu jak i oprogramowania. Czujniki, sterowniki, regulatory są ze sobą połączone w sieć, stosuje się do tego różne okablowanie. Urządzenia nadawczo-odbiorcze wykorzystywane są do komunikacji bezprzewodowej. Wszelkiego rodzaju bramy, mosty, wzmacniacze (ang. *repeatery*), przełączniki czy routery to typowe rozwiązania służące do budowy sieci. Cały ten sprzęt powinien umożliwiać wzajemną komunikację, nadzór, kontrolę, odczyt, gromadzenie i przetwarzanie pomiarów. Służy do tego różnego rodzaju oprogramowanie. Użytkownik końcowy dysponuje najczęściej programami komputerowymi, które prezentują działanie danej inteligentnej instalacji. Same urządzenia powinny działać w oparciu o zaprogramowane w nich instrukcje, programy takie powinny również kontrolować ich pracę i ewentualnie wykrywać wszystkie nieprawidłowości. W celu komunikacji urządzeń pracujących w inteligentnym budynku między sobą, czy też między urządzeniami a użytkownikiem, wykorzystuje się różnego rodzaju protokoły. W początkowych fazach rozwoju systemów BAS, w ramach jednej instalacji mogło występować kilka protokołów. Ich użycie uzależnione mogło być od rodzaju sprzętu, inny protokół do komunikacji, inny do zastosowanej automatyki. Ponadto, różnego rodzaju sprzęt sterujący nie obsługiwał wszystkich funkcji, przez co trzeba było stosować rozwiązania dostarczane przez różnych producentów. Każdy z nich, dysponował najczęściej swoimi rozwiązaniami komunikacyjnymi (różne protokoły).

Wraz z popularyzacją systemów inteligentnych budynków pojawiły się rozwiązania, które umożliwiały budowę instalacji niezależnych od producentów, sprzętu sterującego czy sieciowego. Zjednoczenie i zunifikowanie rozwiązań wpłynęło na szybki rozwój jak i popularyzację automatyki służącej do sterowania budynkiem.

Do popularnych protokołów komunikacyjnych w systemach BAS/BMS zaliczamy:

- BACnet,
- LonTalk,
- PROFIBUS FMS,
- EIB,
- EIBnet
- MODBUS,
- CAN,
- X10,
- P-NET.

3.2.2. Protokół BACnet

BACnet⁵ jest protokołem otwartym i niezależnym od sprzętu umożliwiającym komunikację systemów sterowania i monitorowania. Powstał w 1987 roku, w 1995 roku został zatwierdzony jako standard przez organizację ASHRAE/ANSI, a w 2003 przez ISO [31]. W specyfikacji protokołu określono warstwy komunikacji, odpowiadające modelowi OSI. Działanie protokołu opiera się o przysyłanie komunikatów (zwanymi zapytaniami serwisowymi), a model przypomina strukturę klient-serwer. Komunikacja między urządzeniami odbywa się na zasadzie pytanie-odpowiedź. W opisie protokołu zdefiniowanych jest 35 typów wiadomości odpowiadających 6 usługom (klasom). Przykładowe usługi używane w komunikacji między urządzeniami przedstawia tabela 3.1.

Tab. 3.1 Usługi używane w komunikacji między urządzeniami w budynku [31]

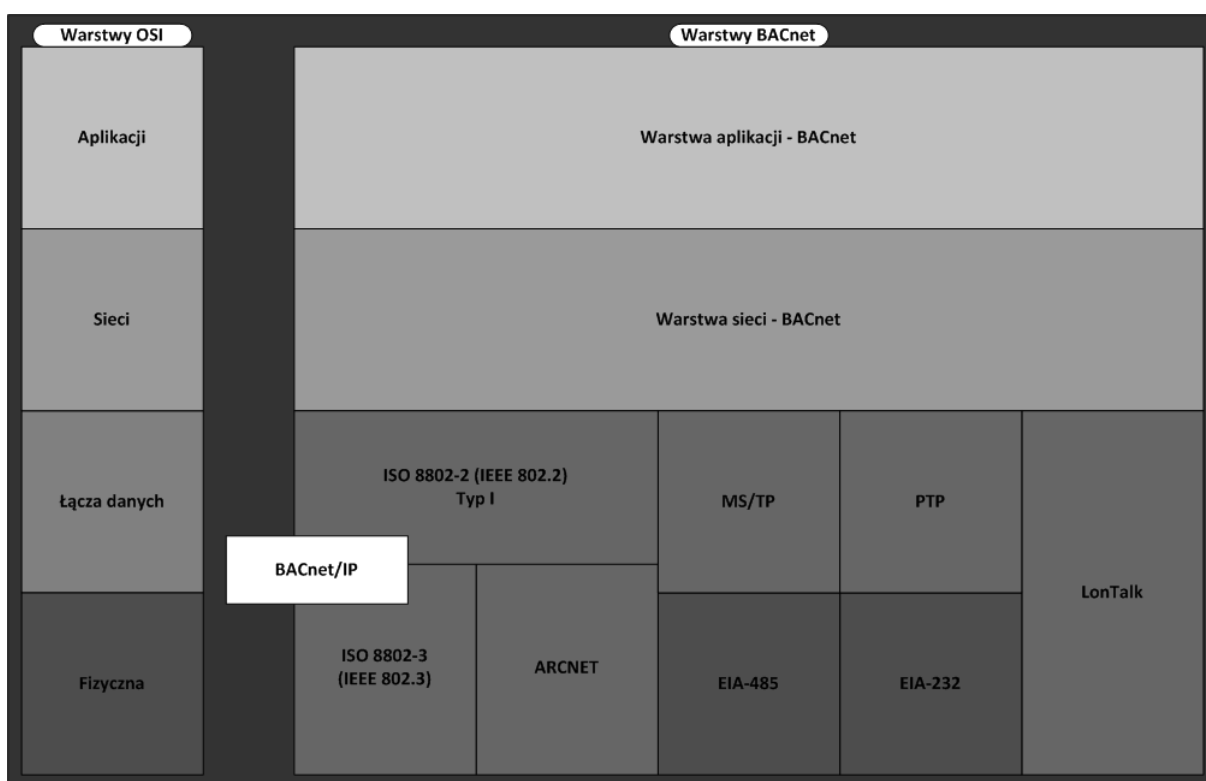
Wykrywanie obiektów i urządzeń	Wymiana danych
Who-Is	Read-Property
I-Am	Write-Property
Who-Has	
I-Have	

Protokół BACnet w obrębie warstwy fizycznej/łącza danych zaleca używanie kilku wariantów połączeń [31]:

- ARCNET,
- Ethernet,
- BACnet/IP,
- Point-To-Point poprzez RS-232,
- Master-Slave/Token-Passing poprzez RS-485,
- LonTalk.

⁵ BACnet (ang. Building Automation and Control Networks) – protokół komunikacji danych stosowany w sieciach inteligentnych budynków

Porównanie protokołu BACnet do modelu ISO/OSI przedstawia rysunek 3.2.



Rys. 3.2 Warstwy protokołu BACnet – porównanie z OSI [21]

3.2.3. Protokół LonTalk

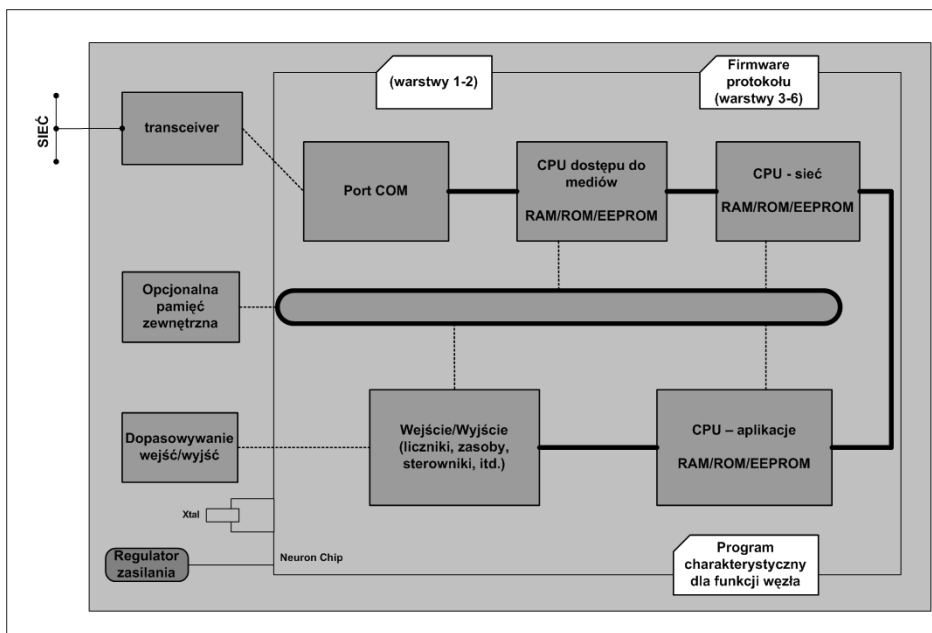
LonTalk jest protokołem opracowanym przez konsorcjum Echelon. Sieć połączeń węzłów może występować w dowolnej topologii i przy użyciu mediów takich jak: skrętka, światłowód, przewód koncentryczny, fale radiowe lub sieć energetyczna [33]. Do łączenia węzłów wykorzystujących różne typy mediów, wykorzystuje się routery (bramy). Najpopularniejszym medium jest skrętka, a sieć zbudowana w jej oparciu może działać z prędkością 78 kbps lub 1,25 Mbps. LonTalk jest częścią platformy LonWorks i powstał w oparciu o model ISO/OSI. Dzięki temu, daje możliwość łatwego łączenia się z innymi systemami. Protokół LonTalk działa na podobnych zasadach co protokół CSMA⁶, jednak ze skróconymi czasami oczekiwania po zaistnieniu kolizji i redukcji komunikatów [21]. W ten sposób zmniejszono ruch w sieci jak i prawdopodobieństwo występowania kolizji.

Działanie sieci LonWork opiera się głównie o wysyłanie komunikatów między dwoma węzłami, przy czym w jednym węźle (np. czujniku temperatury) definiuje się zmienną wyjściową, a w drugim (np. kontrolerze temperatury) zmienną wejściową tego samego typu. Funkcje pierwszego węzła to cykliczne wysyłanie odczytanej temperatury, która następnie jest odczytywana w drugim węźle i w zależności od jej wartości wykonuje się odpowiednia

⁶ ang. CSMA – Carrier Sense Multiple Access (wielokrotny dostęp z wykrywaniem nośnej)

procedura (np. otwarcie/zamknięcie zaworu) [25]. Wszystkie wymiany komunikatów w sieci LonWorks mogą dokonywać się jednocześnie na wielu kanałach. Neuron Chip to procesor opracowany specjalnie do wykonywania operacji w węźle. Procesor wykonuje program, w którym zmiennymi są dane, wykorzystywane do wymiany komunikatów z innymi węzłami [19]. Do obsługi różnego typu mediów procesor wykorzystuje transceivery.

Schemat Neuron Chipa przedstawiono na rysunku 3.3.



Rys. 3.3 Schemat urządzenia LonWorks z układem Neuron Chip [21]

3.2.4. Protokół PROFIBUS

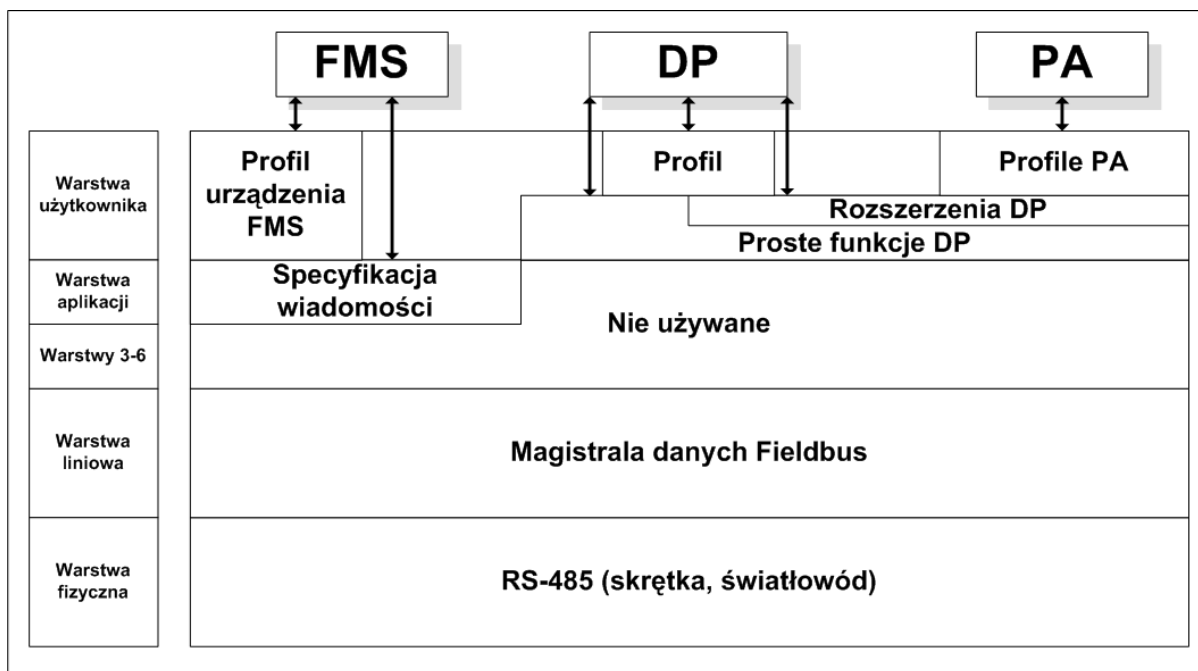
PROFIBUS⁷ jest standardem sieci przemysłowych działających w czasie rzeczywistym. Opracowany w Niemczech przez Ministerstwo Badań i Technologii. Technologia składa się z trzech elementów [34]:

- PROFIBUS-FMS (ang. *Fieldbus Message Specification*) wzorowany na modelu klient-serwer używany do komunikacji między urządzeniami;
- PROFIBUS-DP (ang. *Decentralized Peripherals*) obsługa szybkich zapytań we/wy, podłączania sensorów i aktorów⁸ do urządzeń sterowniczych;
- PROFIBUS-PA (ang. *Process Automation*) używany jest do połączeń urządzeń dostępowych i transceiverów z urządzeniami kontroli dostępu. Zapewnia bezpieczeństwo wewnętrznej transmisji i zasilania linii. Dla tego elementu zdefiniowane są odpowiednie parametry i bloki funkcyjne obsługujące proces technologiczny.

⁷ ang. PROFIBUS – Process Field Bus

⁸ ang. actuator – element wykonawczy w systemach regulacji, który na podstawie sygnału sterującego wypracowuje sygnał wejściowy do obiektu regulacji [36]

Przy budowie protokołu transmisji w technologii PROFIBUS oparto się na modelu ISO/OSI. Warstwa aplikacji korzysta z funkcji komunikacyjnych elementu FMS, posiada podwarstwę LLI (ang. *Lower Layer Interface*) zawierającą funkcje warstw 3-6 OSI. Najniższe warstwy 1 i 2 formułują medium transmisyjne i jego protokół dostępu, sygnały elektryczne interfejsów i usługi warstwy łącza danych. Z poziomu użytkownika dostępne są funkcje zarządzania siecią – FMA (ang. *Fieldbus Management Application*). Architekturę PROFIBUS przedstawiono na rysunku 3.4.



Rys. 3.4 Architektura PROFIBUS wobec modelu ISO/OSI [21]

W standardzie PROFIBUS stosuje się transmisję zgodną ze standardem RS-485. Do budowy sieci wykorzystywana jest skrętka lub światłowód. W sieci może działać maksymalnie 127 stacji. Protokół działa w trybie klient-serwer z komunikacją typu master/slave bądź master/master, przy czym w jednym czasie niezależnie może się komunikować ze sobą wiele urządzeń. Transmisja w sieci jest asynchroniczna w trybie półdupleks. Warstwa fizyczna zapewnia komunikację rozgłoszeniową wiadomości z i bez potwierdzeń. Sterownik udostępnia szereg zmiennych, łańcuchów tekstowych, tablic, rekordów podając ich nazwę. Inne sterowniki mogą odczytywać lub zapisywać zmienne, jeżeli pozwalają na to ustawione prawa dostępu. Możliwa jest również kontrola, przesyłanie programów i zdarzeń [28]. Prędkość transmisji zależna jest od długości linii (tabela 3.2), maksymalna długość jednego odcinka wynosi 1,2 km. Dzięki zastosowaniu wzmacniacza, długość odcinka wzrasta do 4,8 km.

Tab. 3.2 Zależność prędkości transmisji od długości medium transmisyjnego

PROFIBUS-FMS		PROFIBUS-DP	
Prędkość transmisji	Długość segmentu	Prędkość transmisji	Długość segmentu
9,6 kbps	1200 m	1500 kbps	200 m
19,2 kbps	1200 m	12000 kbps	100 m
93,75 kbps	1200 m		
187,5 kbps	1000 m		
500 kbps	400 m		

3.2.5. Protokół EIB

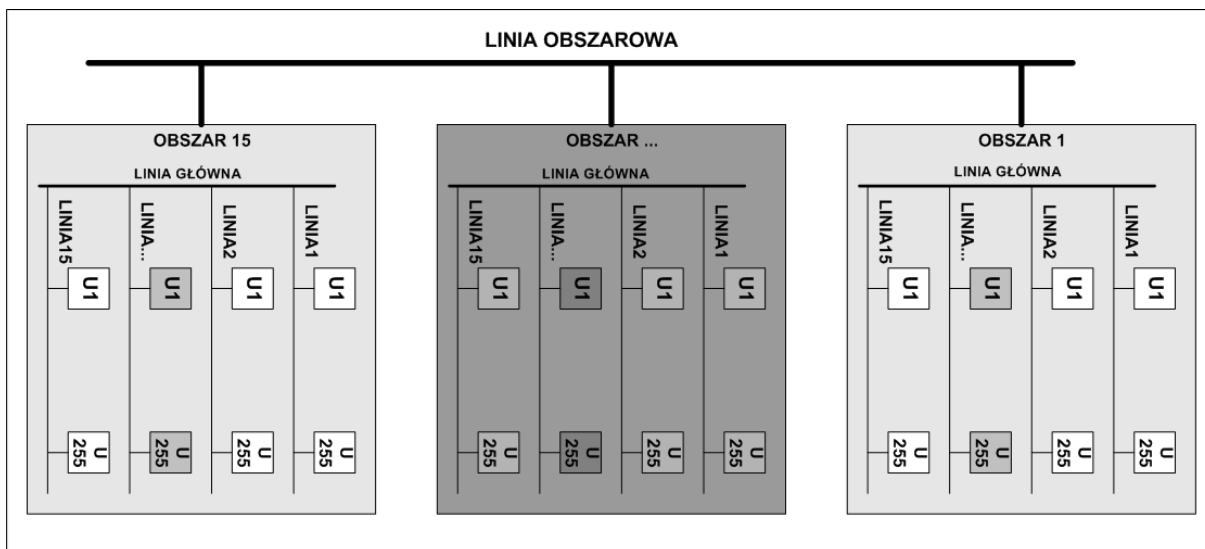
EIB⁹ jest to system instalacji sterującej w budynkach biurowych i mieszkalnych, istniejący jako dyrektywa Unii Europejskiej, ściśle związany z pojęciem „inteligentnego budynku”. Umożliwia on podłączenie do jednej magistrali wszystkich urządzeń pomiaru, regulacji, sterowania, kontroli stanu i nadzoru w danym budynku. Do budowy systemu wykorzystuje się najpowszechniej przewód miedziany „skrętkę”, 2 pary przewodów o przekroju 0,8 mm, przy czym 2 pary stosuje się jako rezerwę. Standard EIB określa użycie tego medium jako EIB.TP (ang. *Twisted Pair*), dopuszcza jednocześnie wykorzystanie elektrycznej linii zasilającej EIB.PL (ang. *Power Line*) lub fale radiowe EIB.RF (ang. *Radio Frequency*) [21].

Magistrala Europejskiej Magistrali Instalacyjnej dopuszcza stosowanie 3 topologii połączeń między urządzeniami: linia, gwiazda, drzewo. W sieci każde urządzenie ma równe prawa dostępu do medium (ang. *Multi Master CSMA/CA*), dzięki czemu różne topologie mogą współpracować w ramach jednej sieci. W zależności od zastosowanego medium, sieć EIB może pracować z różnymi prędkościami. Prędkość transmisji w przypadku EIB.TP wynosi 9600 bps. EIB.PL pracuje z prędkością 1200 bps, przy czym zasięg wynosi 600 m, jest to maksymalna odległość między dwoma urządzeniami, bez stosowania wzmacniacza. Sieć EIB.RF może działać z maksymalną przepustowością 16,384 kbps, zasięg w terenie niezabudowanym wynosi maksymalnie 300 m.

Konstrukcja sieci EIB jest wielopoziomowa i występują w niej różne rodzaje urządzeń. Podstawowym elementem systemu jest linia, do niej podłączone są dane urządzenia (czujniki, przełączniki, itp.). Kilka linii łączy się w tzw. obszar. W ramach danego obszaru wydziela się linie główną, służącą do połączeń między poszczególnymi liniami. Linia obszarowa natomiast, służy do łączenia poszczególnych obszarów, spajając jednocześnie wszystkie urządzenia i linie w jedną sieć. Do jednej linii może być podłączonych maksymalnie 255 urządzeń. Jeden obszar, umożliwia spięcie ze sobą do 15 linii. W związku z tym, w danej

⁹ ang. EIB (European Installation Bus) – Europejska Magistrala Instalacyjna

sieci może maksymalnie pracować jednocześnie 57375 urządzeń ($15 \cdot 15 \cdot 255$). Każda linia zasilana jest prądem 24 V DC, dostarczonym z zasilacza. Zasilacz taki posiada zabezpieczenia napięciowe i prądowe, dzięki temu awaria danego urządzenia (lub zasilacza), nie wpływa na pracę pozostałej części sieci. Schemat sieci przedstawia rysunek 3.5.



Rys. 3.5 Połączenia między urządzeniami, liniami i obszarami w sieci EIB [35]

W dynamicznie rozwijającym się świecie, sprawdzone rozwiązania istniejące długo na rynku potrzebują dostosować się do zmiennych warunków. EIB.net jest rozszerzeniem standardu EIB, dostosowanym do pracy w „sieci”. Istniejące w budynkach instalacje oparte o protokół IP jak i Internet, dają właśnie taką możliwość, implementację systemu EIB w nowoczesnych sieciach. Zarządzanie budynkiem, przy użyciu tej odmiany, umożliwia podłączenie urządzeń automatyki i zarządzania budynkiem bezpośrednio do jednej sieci [21].

Z połączenia standardów organizacji EIB, EHS¹⁰ i BATIBUS¹¹ powstał pierwszy na świecie otwarty standard kontroli urządzeń i zarządzania budynków. Standard KONNEX/KNX jest zatwierdzony przez europejską normę EN50090 – EN13321-1, jak i światowy standard ISO/IEC 14543. Może być używany do kontroli różnych funkcji i aplikacji w domach i budynkach, np. oświetlenie, ogrzewanie, wentylacja, klimatyzacja, rolety i zasłony, monitoring, alarm, systemy bezpieczeństwa... Obsługuje kilka mediów komunikacyjnych: skrętka, instalacja elektryczna, Ethernet i fale radiowe. Otwartość systemu KNX pozwala stosować urządzenia dostarczane przez różnych producentów. Standard KNX opiera się na stosie komunikacyjnym EIB, rozszerzonym o dodatkowe warstwy fizyczne,

¹⁰ EHS (ang. European Home Systems Association) – europejskie stowarzyszenie systemów budynkowych.

¹¹ BatiBUS – pochodzący z Francji standard (NFC 46620), zapewnia 1,2, i 7 warstwę ISO/OSI.

tryby konfiguracyjne jak i możliwości aplikacyjne, pochodzące ze standardów BatiBUS i EHS. KNX może pracować w trzech trybach, dostosowanym do potrzeb użytkowników [22]:

- tryb A – „*Automatic Mode*” – urządzenia automatycznie skonfigurowane, przeznaczone do kupna i instalacji przez użytkownika końcowego,
- tryb E – „*Easy Mode*” – urządzenia pre-programowane, odpowiednie parametry pracy należy dostosować do wymagań użytkownika,
- tryb S – „*System Mode*” – urządzenia tworzone na zamówienie, bez żadnych ustawień domyślnych, muszą być zaprogramowane przez odpowiednich techników.

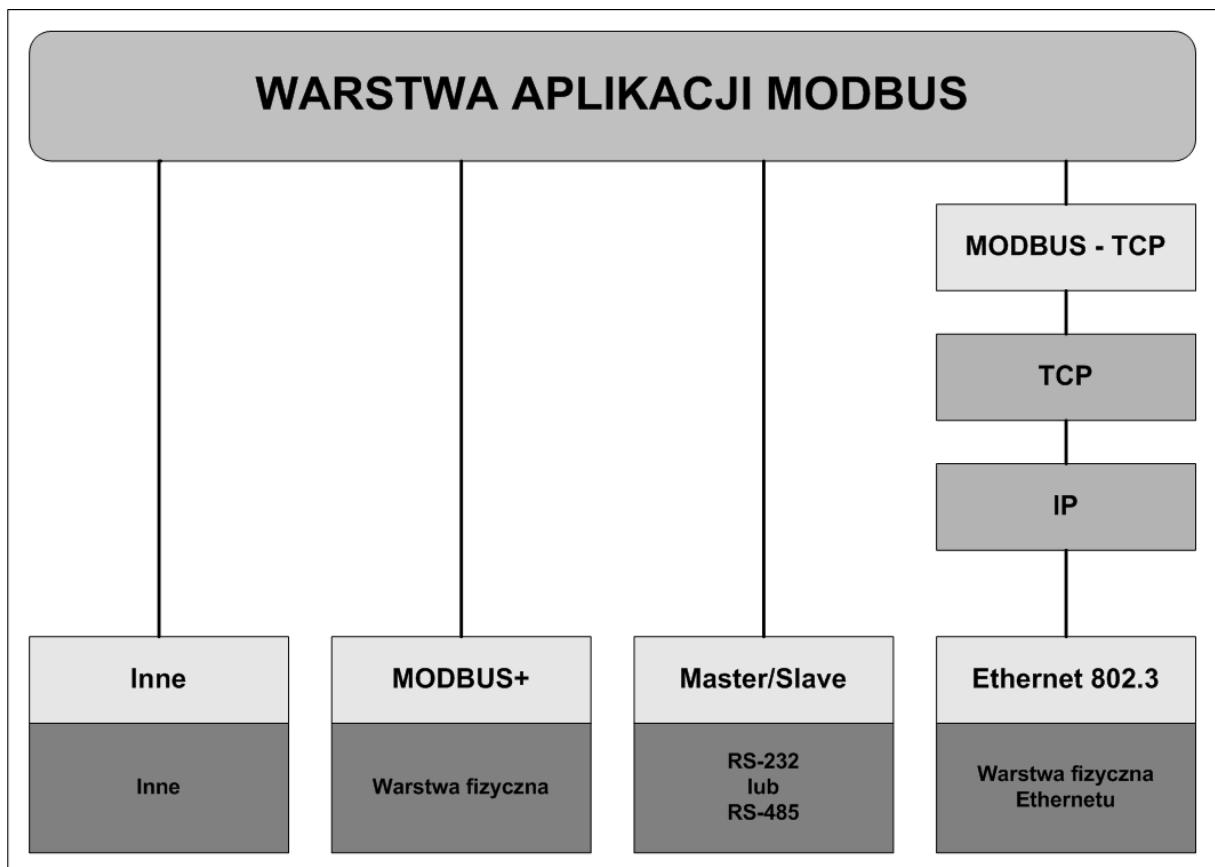
3.2.6. Protokół MODBUS

MODBUS jest protokołem opracowanym przez firmę Modicon. Jego działanie opiera się na zasadzie wymiany komunikatów (komunikacja znakowa), odpowiada warstwie aplikacji modelu OSI. Protokół definiuje PDU (ang. *Protocol Data Unit*), jest to rodzaj pakietu, uniezależniony od danych pochodzących od niższych warstw. Do mapowania pakietu, w celu dostosowania go do różnych sieci lub magistral, PDU może zawierać dodatkowe informacje zapisane w ADU (ang. *Application Data Unit*). Pakiet ADU zawiera adres, kod funkcji i dane (PDU), słowo kontrolne. Urządzenie nadrzędne (ang. *master*) wysyła żądanie do urządzenia podrzędnego (ang. *slave*) i czeka na odpowiedź zwrotną [24]. Ramka ADU jest tworzona przez klienta, który inicjuje transakcję. Kod funkcji wskazuje serwerowi, jaką operację wykonać. Protokół MODBUS ustanawia format żądania zainicjowanego przez klienta. Kod funkcji jest polem o długości jednego bajta. Do dyspozycji jest przedział dziesiętny od 1 do 255, przy czym funkcje o numerach od 128-255 są zarezerwowane (obsługa wyjątków). Kiedy od klienta do serwera zostanie przesłana wiadomość, kod funkcji informuje serwer, jaką operację wykonać. Kod funkcji o numerze „0” jest niepoprawny. Kiedy serwer odpowiada klientowi, używa kodu funkcji (w przypadku prawidłowej transakcji) lub kodu wyjątku (kiedy nastąpi jakiś błąd). Odpowiedzią przy prawidłowej transmisji jest odesłanie przez serwer tego samego kodu, który przysłał klient i danych zawierających odpowiedź. Kiedy nastąpi błąd, serwer odsyła do klienta kod funkcji (ten sam) z najstarszym bitem ustawionym na logiczne „1”. Ramka PDU ma maksymalną długość równą 253 bajty (256 bajtów – bajt adresu serwera – 2 bajty CRC¹²). Ramka ADU w komunikacji szeregowej RS-422/485 to maksymalnie 256 bajtów (PDU + adres + CRC). Dla komunikacji w Ethernetie ramka ADU może mieć maksymalnie 260 bajtów (PDU + 7 bajtów

¹² CRC (ang. Cyclic Redundancy Check – cykliczny kod nadmiarowy) to matematyczna suma kontrolna wykorzystywana do wykrywania uszkodzonych danych binarnych.

MBAP¹³). Implementacja protokołu MODBUS jest możliwa przy użyciu następujących technologii (rysunek 3.6):

- TCP/IP (zarezerwowany port systemowy nr 502),
- szeregowa transmisja asynchroniczna przy użyciu mediów takich jak:
 - przewód RS-232, RS-422/485,
 - przewód optyczny,
 - fale radiowe,
- MODBUS PLUS, szybka sieć przekazywania żetonów (ang. *token*).



Rys. 3.6 Stos komunikacyjny protokołu MODBUS [24]

3.2.7. Protokół CAN

CAN (ang. *Controller Area Network*) jest magistralą opracowaną przez firmę BOSCH, zaprojektowaną do sterowania układów pomiarowych i wykonawczych w samochodach. Zgodność transmisji danych z modelem ISO/OSI zaowocowało implementacją w innych dziedzinach automatyki i sterowania, na przykład w sterowaniu ruchem drogowym, aparaturze medycznej i automatyzacji budynków [21]. Szeregowa transmisja asynchroniczna

¹³ MBAP (ang. MODBUS Application Protocol) – protokół aplikacji MODBUS

W 1993 protokół CAN stał się standardem ISO 11898-1, zawiera opis warstwy łącza danych modelu referencyjnego. CAN zapewnia dwie usługi komunikacyjne: przesyłanie wiadomości (transmisja ramek) i żądanie wiadomości (zdalne żądanie transmisji). Pozostałe usługi takie jak sygnalizacja błędów czy automatyczna retransmisja uszkodzonych ramek, są przezroczyste dla użytkownika. Układ scalony CAN ma zaimplementowane odpowiednie algorytmy, które obsługują błędy.

Protokół CAN opiera się o szeregową transmisję asynchroniczną. Sieć CAN posiada hierarchię ang. *multi-master*, dzięki czemu możliwe jest zachowanie działania sieci w przypadku awarii jej części. Przesyłanie wiadomości sieci odbywa się jednocześnie do wszystkich urządzeń. Nadawca rozgłasza informację, a następnie odbiorca odczytuje wiadomość i decyduje, czy jest dla niego istotna. Rozwiązanie takie zapewnia integralność danych, ponieważ wszystkie urządzenia używają takich samych informacji [30].

W sieciach CAN stosuje się najczęściej topologię magistrali (szyny), przy czym możliwa jest praca w topologii gwiazdy (łączenie węzłów). Przesyłane wiadomości mają wielkość ośmiu bajtów, chronione algorytmem CRC-15, podczas transmisji możliwe jest wykrycie kolejno do pięciu błędnych bitów. Prędkość transmisji w sieci CAN wynosi maksymalnie 1 Mbps, na odcinku do 40 m. Prędkość transmisji maleje wraz z odległością, na odcinku 500 m można uzyskać prędkość 125 kbps. Do przesyłania wiadomości w sieciach CAN zastosować można skrętkę ekranowaną, światłowody, sieć energetyczną, podczerwień lub fale radiowe. W zależności od wersji protokołu wyróżniamy: 11-bitowy identyfikator (CAN 2.0A) i 29-bitowy (CAN 2.0B). Ramka danych CAN składa się z siedmiu pól – początku, arbitracji, sterującego, danych, sumy kontrolnej, potwierdzenia i końca [4]. Identyfikator komunikatu (położony w polu arbitracji) określa priorytet dostępu do magistrali – im mniejsza wartość liczbowa, tym priorytet większy. Zaletą magistrali CAN jest to, że identyfikator nie jest przypisany do urządzenia, lecz do komunikatu. Dzięki takiemu rozwiązaniu liczba urządzeń w systemie nie jest ograniczona liczbą adresów. Dostęp do magistrali jest przyznawany poprzez badanie stan magistrali przez wszystkie stacje, czekając na możliwość wysłania własnego komunikatu. Konflikty wynikające z prób równoczesnego nadawania przez kilka stacji są rozwiązywane w początkowej fazie transmisji w trakcie wysyłania pola arbitracji. Dostęp do łącza otrzyma stacja o niższym numerze identyfikacyjnym.

3.2.8. Protokół CANopen

W 1995 roku opracowano protokół CANopen, który rozszerza CAN o wyższe warstwy modelu referencyjnego ISO/OSI (od warstwy sieci włącznie). Składa się ze schematu adresowania, kilku mniejszych protokołów komunikacyjnych oraz warstwy aplikacji zdefiniowanej przez profile sprzętowe. Protokoły komunikacyjne obejmują zarządzanie siecią, kontrolę sprzętu i komunikacji między węzłami. CANopen posiada prostą warstwę transportową, która zajmuje się segmentacją i ponownym składaniem wiadomości.

Każde urządzenie pracujące w sieci CANopen musi mieć zaimplementowany w swoim oprogramowaniu kontrolnym pewien zestaw cech [3]:

- Jednostka komunikacyjna (ang. *communication unit*) z zaimplementowanymi protokołami służącymi do przesyłania wiadomości do innych węzłów w sieci.
- Uruchamianie i restartowanie urządzenia kontrolowane jest przez aparat stanów (ang. *state machine*). Wyróżniamy następujące stany – inicjalizacja, przed-operacyjny, operacyjny, zatrzymanie.
- Słownik obiektów (ang. *object dictionary*) jest tablicą zmiennych o 16-bitowym indeksie. Każda zmienna może mieć dodatkowy 8-bitowy subindeks. Zmienna używana jest do konfigurowania urządzenia, lub do odzwierciedlania jego stanu (np. wynik pomiaru).
- Programowa część urządzenia wykonuje wymagane funkcje urządzenia, po tym jak urządzenie zostanie ustawione w stan operacyjny. Program konfigurowany jest przez zmienne w słowniku obiektów, a dane przesyłane i odbierane są poprzez warstwę komunikacyjną.

Urządzenia CANopen posiadają zdefiniowany słownik obiektów. Wpis do słownika zbudowany jest z:

- **indeksu**, 16-bitowy adres obiektu w słowniku,
- **nazwy obiektu**, symboliczny typ obiektu we wpisie słownikowym (tablica, rekord, zmienna),
- **nazwy**, łańcuch tekstowy opisujący dany wpis do słownika,
- **typu**, określa typ danych konkretnej zmiennej,
- **atrybutu**, określa prawa dostępu do wpisu (odczyt/zapis, tylko odczyt, tylko zapis, stała tylko do odczytu).

Słownik obiektów może zostać wgrany do urządzenia przy pomocy odpowiednio sformatowanego pliku *.ini.

CANbus, warstwa fizyczna protokołu CANopen może przysyłać wyłącznie krótkie pakiety, zawierające 11-bitowy identyfikator id, bit zdalnego żądania transmisji (RTR) i do ośmiu bajtów danych. Standard CANopen dzieli 11-bitową ramkę identyfikatora na 4-bitowy kod funkcji i 7-bitowy identyfikator węzła sieci. Liczba urządzeń w sieci ograniczona jest przez to do 127 (CAN 2.0 B dopuszcza 29-bitowe ramki identyfikacyjne). W przypadku wystąpienia kolizji, arbitracja powoduje przepuszczenie w pierwszej kolejności ramki z najmniejszym identyfikatorem.

Do przesyłania wiadomości pomiędzy węzłami w sieci CANopen wykorzystuje się trzy modele [32]:

- master/slave – Jeden węzeł oznaczony jako master wysyła bądź żąda danych od węzłów slave.
- klient/serwer – Klient SDO¹⁴ wysyła dane do serwera (indeks i subindeks katalogu obiektu), który odpowiada jednym bądź więcej pakietami zawierającymi żądane dane (zawartość katalogu obiektów pod wskazanym indeksem).
- producent/konsument – Model używany w protokołach Heartbeat¹⁵ i ochrony węzła¹⁶. Producent wysyła dane do konsumenta bez wcześniejszego żądania albo producent wysyła dane na żądanie konsumenta.

Standard CANopen obejmuje pięć warstw modelu referencyjnego ISO/OSI, przez co wyróżnić w nim można kilka protokołów. Każdy związany bezpośrednio z daną warstwą czy obsługą jakiejś usługi lub operacji. Charakterystyka poszczególnych protokołów przedstawiona jest poniżej.

Protokoły zarządzania siecią (ang. *Network management (NMT)*) używane są do zmiany stanu urządzeń (uruchomienia lub zatrzymania urządzenia), wykrywania wzbudzenia urządzeń zdalnych i obsługi błędów. Do zarządzania siecią stosuje się dwa typy protokołów. Pierwszy, protokół kontroli modułów (ang. *module control*) jest używany przez urządzenie master do zmiany stanów urządzeń. Drugi (ang. *heartbeat*¹⁴), do sprawdzania czy urządzenia pracujące w sieci są aktywne [5].

Protokół obsługi danych obiektów (ang. *Service Data Object (SDO)*) używany jest do ustawiania i odczytywania wartości z katalogu obiektu zdalnego urządzenia. Urządzenie, którego katalog jest odczytywany nazywany jest serwerem, a urządzenie mające dostęp do

¹⁴ SDO (ang. Service Data Object) – protokół używany do ustawiania i odczytywania wartości z katalogu obiektów urządzenia zdalnego.

¹⁵ ang. Heartbeat – monitorowanie sieci i węzłów w celu sprawdzenia czy są aktywne, urządzenia (najczęściej slave) przysyłają cyklicznie wiadomość z kodem 1110 i swoim identyfikatorem.

¹⁶ ang. Node Guarding – model typu reakcja/odpowiedź używany do monitorowania urządzeń slave, wyparty przez protokół Heartbeat.

urządzenia zdalnego klientem. Komunikacja inicjowana jest przez klienta. W terminologii CANopen, komunikacja widziana jest od strony serwera, przez co czytanie z katalogu obiektu nazywa się operacją wysyłania (ang. *upload*), natomiast zapis do katalogu jest operacją pobierania (ang. *download*). Wartości katalogu obiektu mogą być większe niż ramka CAN (limit długości osiem bajtów), protokół SDO posiada zaimplementowane segmentowanie i składanie większych wiadomości [7].

Protokół przetwarzania danych obiektów (ang. *Process Data Object (PDO)*) używany jest do wymiany w czasie rzeczywistym danych pomiędzy węzłami sieci. Ramka PDO może zawierać do ośmiu bajtów danych (mapowanie do ramki CAN), które są transmitowane do lub z urządzenia. Ramka transmisji danych (ang. *transmit PDO – TPDO*) umożliwia czytania danych z urządzeń. Ramka odbierania danych (ang. *receive PDO – RPDO*) pozwala na wysyłanie danych do urządzeń. Ramki PDO mogą być wysyłane synchronicznie lub nie. W pierwszym przypadku, ramka wysyłana jest po wysłaniu wiadomości SYNC, a w drugim na wypadek zewnętrznego lub wewnętrznego wzbudzenia (np. przesłanie specjalnie spreparowanej ramki TPDO) [6].

Protokół synchronizacji obiektów (ang. *Synchronization Object (SYNC)*) stosowany jest, kiedy urządzenie synchronizujące wysyła odpowiedni sygnał do urządzenia synchronizowanego, po tym jak „adresat” otrzyma sygnał, obydwa urządzenia wykonują szereg operacji. Czujniki/sensory mogą dostroić się do wzorcowych zmiennych procesowych i że aktry mogą ustalić swoje działanie w zharmonizowany sposób [8].

Protokół znaczników czasowych (ang. *Time Stamp Object (TIME)*) definiuje znacznik czasowy, który odpowiada całkowitemu czasowi, wyrażonemu w milisekundach i dniach, od północy dnia 1 stycznia 1984 roku. Stempel czasu reprezentowany jest przez sekwencję bitów długości sześciu bajtów. W niektórych sieciach, gdzie pracują urządzenia wymagające wysokiej dokładności w synchronizacji, stosuje się znaczniki czasowe wysokiej rozdzielczości. Taki stempel kodowany jest liczbą czterech bajtów bez znaku (ang. *unsigned32*) z rozdzielczością 1 μ s (licznik czasu zeruje się co 72 minuty) [8].

Protokół awaryjny (ang. *Emergency Object (EMCY)*) odpowiada za przesyłanie wiadomości awaryjnych, które wywoływane są przez pojawienie się błędu krytycznego w urządzeniu. Informacja o takim zdarzeniu przesyłana jest do pozostałych urządzeń z wysokim priorytetem. Dzięki temu urządzenia gotowe są na wystąpienie ewentualnych przerw. Wiadomość awaryjna może być wysłana tylko raz, o ile nie wystąpią żadne nowe (inne) błędy krytyczne [8].

Prędkość transmisji, maksymalna odległość i czas opóźnień pracy urządzeń w sieci CANopen przedstawia tabela 3.3.

Tab. 3.3 Charakterystyka pracy sieci CANopen [2]

Prędkość transmisji	Maksymalna odległość	Maksymalne opóźnienie
1 Mbps	25 m	125 ns
800 kbps	50 m	125 ns
500 kbps	100 m	125 ns
250 kbps	250 m	250 ns
125 kbps	500 m	500 ns
50 kbps	1000 m	1,25 μ s
20 kbps	2500 m	3,125 μ s
10 kbps	5000 m	6,25 μ s

3.2.9. Protokół X10

System protokołu X10 jest otwartym, międzynarodowym standardem wykorzystywanym w domowej automatyce. Do komunikacji wykorzystuje się przede wszystkim istniejące w budynkach instalacje sieci elektrycznej. Ponadto standard dopuszcza komunikację opartą o fale radiowe. X10 pomimo niskiej prędkości transmisji, jest bardzo popularny, ze względu na bardzo niski koszt urządzeń.

Domowa sieć elektryczna, która zasila oświetlenie i wszystkie urządzenia RTV/AGD, jest używana do przesyłania cyfrowych sygnałów pomiędzy urządzeniami X10. Cyfrowa informacja jest kodowana impulsem o częstotliwości 120 kHz, który jest przesyłany w trakcie przejścia przez zero napięcia sieci elektrycznej [38]. Na jedno przejście przypada wyłącznie jeden bit. Informacje przesyłane między urządzeniami pracującymi w sieci X10 składają się z adresu i komendy wydanej przez kontroler do urządzenia odbiorczego. Bardziej zaawansowane kontrolery potrafią odpytywać kilka urządzeń, żądając w odpowiedzi przesłania statusu. Status ten może być postaci „włączony” lub „wyłączony”, aktualny poziom ściemnienia, aktualną temperaturę lub poziom innego typu sensora. Urządzenia podłącza się do gniazdka elektrycznego w ścianie.

Niezależnie od tego czy do transmisji stosuje się sieć elektryczną, czy komunikację radiową protokół X10 zakłada przesyłanie wiadomości w postaci:

- czterech bitów identyfikujących budynek (ang. *house code*),
- czterech bitów identyfikujących urządzenie,
- czterech bitów określających funkcję.

Dla wygody użytkowników konfigurujących system, za kod budynku przypisuje się literę od A do P a za kod urządzenia liczbę od 1 do 16. W trakcie instalacji systemu, każde urządzenie jest konfigurowane aby rozpoznać 256 możliwych adresów (16 kodów budynku *

16 kodów urządzeń). Dzięki dwustopniowemu adresowaniu, urządzenia mogą odpowiadać na wywołania skierowane bezpośrednio do niego, bądź do przypisanej mu grupy urządzeń (rozpoznawanych poprzez kod budynku). Protokół X10 umożliwia także wysłanie komendy „zbiorowej” na którą odpowiedzą wszystkie urządzenia zainstalowane w danym budynku. Zgaszenie światła w danym pokoju wymaga wysłania wiadomości w postaci – wybierz kod pokoju (np. P1) i „wyłącz”. Kiedy zaistnieje potrzeba ściemnienia światła w biurach na drugim piętrze, należy użyć takiego komunikatu – wybierz kod biura B1, B2, B3 i „ściemnij”. W jednym budynku można zastosować więcej niż jeden kod budynku, dzięki czemu można pogrupować odpowiednie urządzenia i sterować nimi niezależnie. Wysłanie komunikatu „zgaś wszystkie światła” spowoduje wyłączenie oświetlenia wyłącznie w pomieszczeniach przypisanych do jednego kodu budynku.

W sieci prądu przemiennego, binarne „1” reprezentowane jest przez impuls częstotliwości 120 kHz, trwający 1 milisekundę, w punkcie przejścia przez zero, po czym następuje przerwa w impulsie. Binarne zero natomiast, przez brak impulsu, po czym następuje impuls. Urządzenia oczekują sześć milisekund na odebranie impulsu. Każda ramka w sieci X10 przesyłana jest dwukrotnie, w celu uniknięcia fałszywych sygnałów. Podczas przesyłania komunikatów sieci X10 każda ramka rozpoczyna się od kodu „1110”. Następnie występuje kod budynku, a po nim kod funkcyjny. Kod funkcyjny może zawierać identyfikator urządzenia albo numer funkcji, wybór dokonywany jest przez ostatni bit ramki (0 – numer urządzenia, 1 – numer funkcji). Ramki danych muszą być oddzielane przez 6 przekroczeń zera „000000”, kiedy zmieniają się kolejne adresy, kolejne funkcje, bądź przesyłane są kolejno najpierw adres a potem funkcja [27].

W celu zapewnienia działania urządzeń bezprzewodowych, standard X10 umożliwia komunikację radiową. Przesyłane sygnały mają budowę podobną do tych pracujących w sieci elektrycznej. Częstotliwość pracy wynosi odpowiednio 310 MHz w Stanach Zjednoczonych i 433 MHz w Europie. Konwersji pakietów sygnału radiowego na typowy pakiet X10 dokonuje odbiornik radiowy. Do urządzeń bezprzewodowych pracujących w standardzie X10 można zaliczyć [37]:

- kamery wideo,
- przełączniki oświetlenia,
- przekaźniki i wzmacniacze sygnału radiowego,
- bezprzewodowa transmisja audio/video (telewizja/muzyka w każdym punkcie sieci),
- sensory otwarcia drzwi/okien,
- sensory ruchu.

Zaletą systemu X10 jest jego niski koszt wdrożenia i możliwość błyskawicznej instalacji bez konieczności kładzenia dodatkowych przewodów. Standard ten posiada też wady. W nowoczesnych budynkach działa różny typ urządzeń elektrycznych i elektronicznych. Sprzęt AGD/RTV może pobierać ponad 400 W przez co wprowadza do sieci elektrycznej zakłócenia, które mogą tłumić sygnały sterujące X10. Montowane powszechnie żarówki energooszczędne pobierają mniejszą moc niż wymagać mogą sterowniki oświetlenia. Komputery, filtry antyprzepięciowe również mogą doprowadzić do wytłumienia przesyłanych w sieci rozkazów. Protokół X10 jest wolny, potrzeba około $\frac{3}{4}$ sekundy na przesłanie adresu i funkcji (dublowanie ramek, rozdzielanie kolejnych ramek). Szczególnie odczuwalne jest powolne działanie sieci w przypadku urządzeń sterowanych przy pomocy komputera.

4. Technologia 1-Wire

4.1. Wprowadzenie

1-Wire to magistrala komunikacyjna, opracowana przez firmę Dallas Semiconductor, zapewniająca przesyłanie z niską prędkością sygnałów i zasilania jednym przewodem (wymagany jest jeszcze przewód zerowy). Technologia 1-Wire jest podobna do I²C¹⁷ z tą różnicą, że operuje z mniejszą prędkością, większym zasięgiem i niższym kosztem. Wykorzystywana jest do komunikowania z tanimi urządzeniami, takimi jak: czujniki temperatury, układy pamięciowe, monitory pracy baterii. Możliwość pracy przy użyciu jednej pary przewodów jest możliwe, dzięki wyposażeniu układów w kondensator o pojemności 800 pF, który dostarcza zasilanie z linii danych. Układy pracujące w technologii 1-Wire występują w różnych postaciach:

- przeznaczone do montażu na płytkach drukowanych (TO-92, TSOC, SOT-223 itp.),
- w metalowych obudowach przypominających baterie zegarkowe (urządzenia iButton).

Łatwość implementacji i powszechność technologii, pozwala na montaż aparatów bez specjalnego kontrolera. W takim przypadku urządzenia 1-Wire stanowią element jakiegoś układu, na przykład montaż kontrolerów pracy baterii w akumulatorach do laptopów czy telefonów komórkowych. Do połączenia instrumentów 1-Wire wykorzystuje się przewody lub skrętkę połączone wtykami modularnymi (RJ11, popularne w telekomunikacji). Urządzenia montowane są w gniazdach, wlutowane w płytkę drukowaną lub przymocowane bezpośrednio do monitorowanego obiektu. Wszelkiego rodzaju układy 1-Wire mogą

¹⁷ ang. Inter-Integrated Circuit – pośredniczący między układami scalonymi, szeregową dwukierunkową magistralą wykorzystywaną w urządzeniach elektrycznych.

pracować niezależnie połączone ze sobą równolegle, jednym przewodem. Takie wiązanie określane jest mianem miniaturowej sieci, nazywanej MicroLan. Do współpracy z komputerem PC wykorzystuje się odpowiednie konwertery (wyposażone np. w port USB lub RS-232), wyposażone w odpowiedni interfejs tłumaczący sygnały 1-Wire.

Technologia iButton, wykorzystuje protokół 1-Wire, przy czym do połączenia urządzeń (miniaturowych układów pamięciowych) nie stosuje się przewodów, wymagany jest jedynie powierzchniowe zwarcie „pastylki” z gniazdem. Pobieranie przez układy iButton zasilania bezpośrednio z gniazda, określa się „zasilaniem pasożytniczym” (ang. *parasite power*). Do identyfikacji wszystkich aparatów 1-Wire wykorzystuje się unikalny 64 bitowy kod przydzielany na etapie produkcji. Dzięki temu, powszechne jest stosowanie technologii 1-Wire (szczególnie kluczy typu iButton) w systemach alarmowych, autoryzacji, kontroli dostępu.

4.2. Opis działania układów 1-Wire

4.2.1. Zasilanie

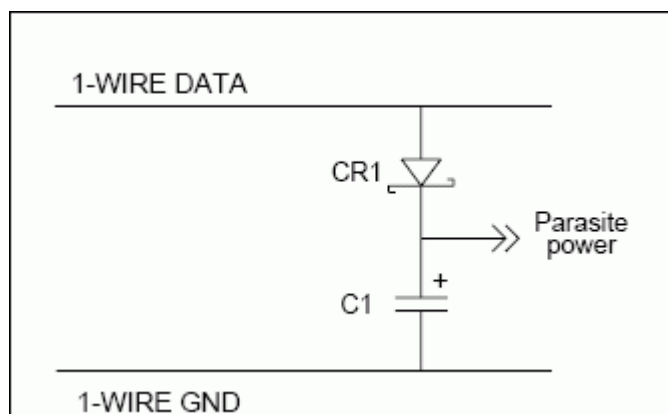
Układy 1-Wire pobierają zasilanie i przesyłają komunikaty używając tego samej linii, jest to rozwiązanie ekonomiczne. Kiedy do danego węzła zostają podłączone kolejne urządzenia pobierające zasilanie, ta sama linia może dostarczyć im zasilania kilkoma sposobami. Przy pomocy linii 1-Wire, w niektórych przypadkach wymagana może być zasilanie dodatkowych elementów, np. obsługa diody LED¹⁸ lub zasilanie czujnika ciśnienia. Poprzez linię komunikacyjną, biorąc dodatkowo pod uwagę zapotrzebowanie na energię, czas przez jaki ma być dostarczona i odległość od urządzenia nadrzędnego, można dostarczyć zasilanie różnymi sposobami.

Wykorzystanie dostępnej energii pomiędzy napięciem 3,5 V a 5 V do zasilania układów. Układy 1-Wire pracują z napięciem dostarczonym na poziomie 3 V, przy czym napięcie magistrali zapewnia do wykorzystania zapas pomiędzy 3,5 V a 5 V. W takim trybie, komunikacja 1-Wire odbywa się poniżej poziomu 3,5 V, dostarczanie energii powyżej tej wartości. Na przykład, gdy układ DS2406 przechodzi w stan niski, następuje zasilanie diody LED a napięcie na linii jest w przybliżeniu równe 3,5 V. Kiedy wyjście przełącznika zostaje wyłączone, dioda gaśnie i magistrala powraca do poziomu 5 V. W przypadku kiedy dioda jest równolegle podłączona do magistrali, jej napięcie będzie utrzymywane kiedy magistrala pozostaje w stanie wysokim. Przy stosowaniu zasilania wykorzystującego przechodzenie

¹⁸ LED, ang. Light Emitting Diode – dioda elektroluminescencyjna.

napięcia pomiędzy poziomami, należy wziąć pod uwagę prąd ładowania płynący przez uziemienie.

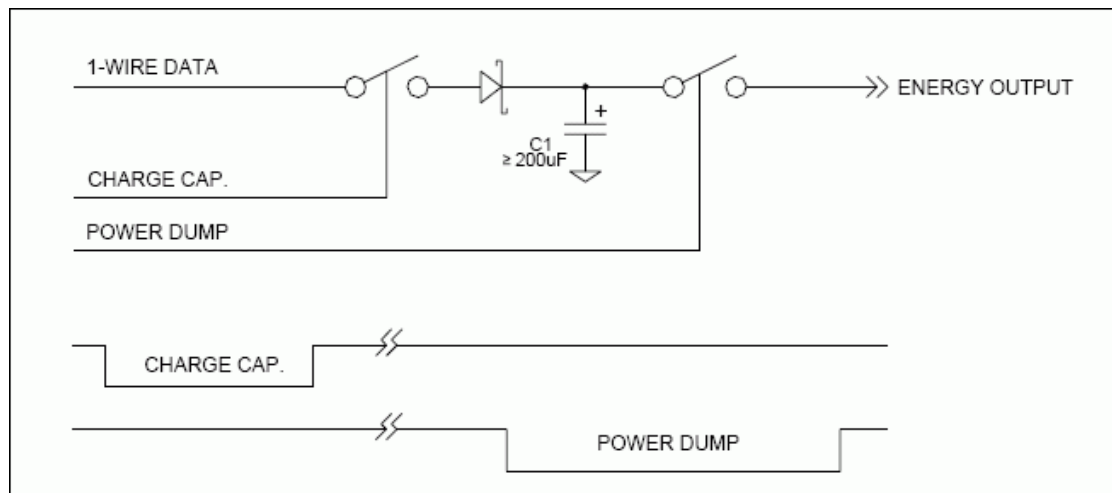
Przekazywanie ładunku do kondensatora przez diodę Shottky'ego. W niektórych zastosowaniach konieczne może stać się zastosowanie zestawu – kondensatora i diody Shottky'ego – w celu dostarczenia zapasu energii. Przykładowy schemat przedstawia rysunek 4.1.



Rys. 4.1 Wykorzystanie kondensatora wraz z diodą Shottky'ego do zasilania [23]

Przykładem wykorzystania tej techniki może być użycie funkcji licznika układu DS2423 do pomiaru prędkości wiatru, gdzie układ korzysta z diody BAT54S i ceramicznego kondensatora o pojemności $0,01\ \mu\text{F}$. W trakcie jałowego trybu komunikacji, magistrala 1-Wire jest na poziomie napięcia 5 V a obwód „kradnie” energię z linii w celu naładowania kondensatora i zasilenia układu. Jest to wykorzystanie techniki „pasożytniczej” (ang. *parasite power*), używanej wewnątrz każdego układu 1-Wire, w celu zapewnienia własnej energii do pracy. Pojemność kondensatora jest uzależniona od zapotrzebowania na prąd i czasu, w jakim napięcie ma być utrzymane na zadanym poziomie. Technika ta jest prosta i niskobudżetowa, jednak ogranicza zasięg i zdolność działania sieci MicroLan. W przypadku, gdy na magistrali występują układy wykorzystujące „pasożytnicze” zasilanie, należy wziąć pod uwagę, że w czasie pracy kondensator jest zwarty, co powoduje zwarcie magistrali i jej nieużyteczność. Do czasu, kiedy kondensator nie zostanie naładowany powyżej 3,5 V, po magistrali 1-Wire nie może przebiegać żadna komunikacja.

Warunkowe dostarczanie energii pod kontrolą urządzenia Master. Na rysunku 4.2 przedstawiono schemat, na którym pomiędzy dwoma adresowalnymi przełącznikami, kontrolowanymi przez układ nadrzędny, umieszczono półokresowy prostownik.



Rys. 4.2 Zasilanie z magistrali 1-Wire, wydzielone dwoma przełącznikami [23]

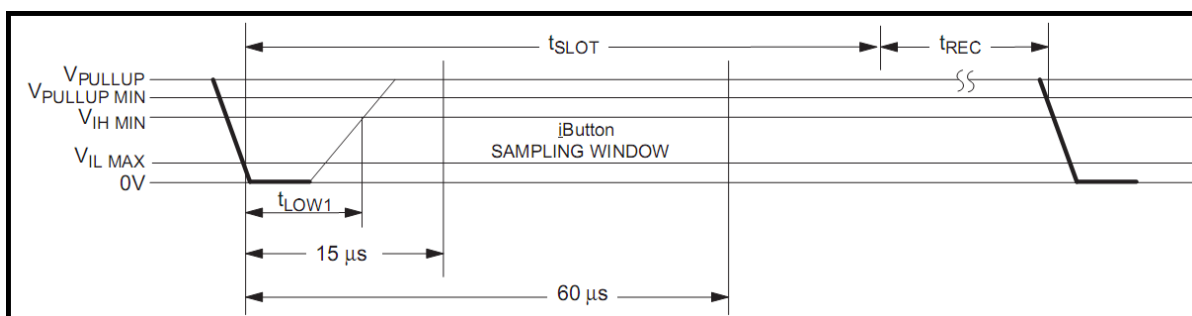
Kiedy przełącznik wejściowy jest zamknięty, kondensator jest zasilany z linii DATA magistrali 1-Wire na tej samej zasadzie, jak na rysunku 4.1. Kiedy przełącznik jest otwarty, kondensator i jego ładunek jest odizolowany od sieci 1-Wire, co pozwala na normalną komunikację, bez obawy o rozładowanie kondensatora czy zwarcie magistrali. W sytuacji, kiedy potrzebna jest nagromadzona energia, przełącznik wyjściowy zostaje zamknięty a kondensator jest rozładowywany i dostarcza wymaganego zasilania. W miejsce kondensatora, przy takim schemacie połączeń, można umieścić baterię akumulatorową. Istotną kwestią jest izolacja elementu gromadzącego energię od magistrali 1-Wire, aby awaria układu nie uszkodziła całej sieci. Kontrolę nad przepływem prądów w sieci powinien w całości sprawować układ master.

4.2.2. Sygnały czasowe

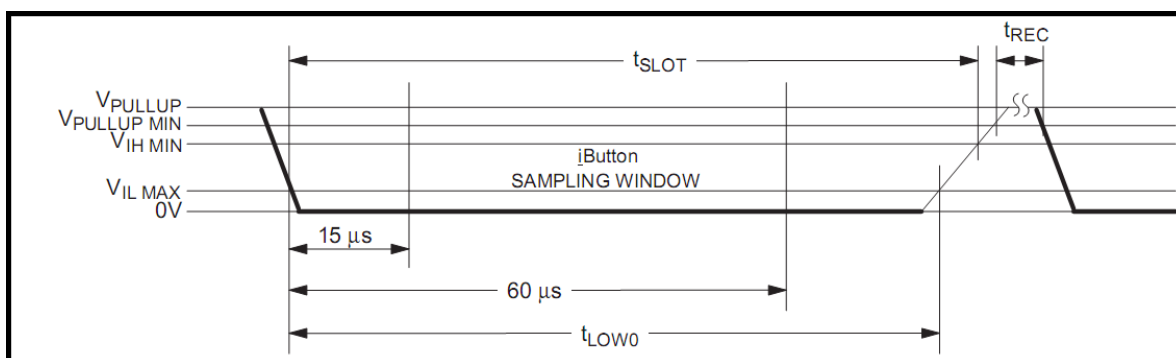
Układy 1-Wire pracują w trybie asynchronicznym i półdupleksowym, przez co do poprawnej komunikacji nie jest wymagana synchronizacja urządzeń, a jedynie zachowanie określonych w standardzie odstępów czasowych determinujących poszczególne sygnały. W celu rozpoznania początku i końca kolejnych bitów informacji, urządzenia 1-Wire reagują na opadanie lub narastanie zbocza napięcia.

Szczelina czasowa zapisu służy do sterowania pracą układów 1-Wire. Wszystkie operacje związane z działaniem urządzeń 1-Wire są zdefiniowane przez szczeliny czasowe (ang. *time slot*). W celu zapewnienia maksimum sprawności działania, urządzenia slave odczytują stan linii danych w środku szczeliny czasowej. Z definicji, aktywna część szczeliny czasowej 1-Wire (t_{SLOT}) trwa 60 μs . W nominalnych warunkach układ 1-Wire zacznie próbkować 30 μs po opadnięciu zbocza rozpoczynającego szczelinę. Tolerowany jest zakres między 15 a 60 μs . Oznacza to, że właściwe próbkowanie może wystąpić w dowolnym czasie

między 15-stą a 60-tą μs . Podczas tej ramki czasowej napięcie na linii danych musi pozostać poniżej poziomu $V_{IL\text{ MAX}}$ lub powyżej $V_{IH\text{ MIN}}$. Czas trwania niskiego stanu magistrali przy zapisywaniu „1” (t_{LOW1}) musi być krótszy niż 15 μs ; do zapisu „0” czas trwania impulsu (t_{LOW0}) musi trwać przynajmniej 60 μs (przez ten czas linia danych jest zwarta do masy). Możliwe jest wydłużenie czasu trwania szczeliny czasowej ponad 60 μs . Ośiem kolejnych szczelin utrzymujących niski stan magistrali jest rozpoznawane jako reset ($8 \cdot 60 = 480\text{ }\mu\text{s}$). Ogranicza to czas trwania próbkowania maksymalnie do 120 μs . Pozwala to uniknąć sytuacji, w której cała szczelina zostanie zinterpretowana jako reset ($8 \cdot 15 = 120\text{ }\mu\text{s}$). Pod koniec szczeliny czasowej urządzenia 1-Wire potrzebują czasu na regenerację stanu magistrali (podniesienie napięcia magistrali do poziomu V_{PULLUP} . Okres t_{REC} trwa minimum 1 μs , po czym następuje opadanie zbocza, sygnalizujące nadejście kolejnego bitu. Szeroka tolerancja szczeliny czasowej pozwala spełnić wymogi komunikacji 1-Wire nawet wolnym mikroprocesorom. Na rysunkach 4.3 oraz 4.4 przedstawiono schematy szczelin czasowych zapisu „1” i „0”.



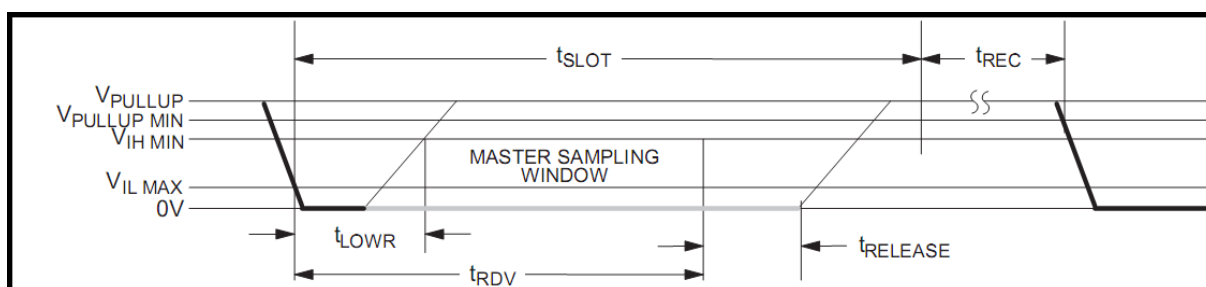
Rys. 4.3 Szczelina czasowa zapisu „1” na magistrali 1-Wire [11]



Rys. 4.4 Szczelina czasowa zapisu „0” na magistrali 1-Wire [11]

Szczelina czasowa odczytu zbiera dane przekazywane przez układy podłączone do magistrali. Funkcje (komendy) i dane są przesyłane o urządzeń 1-Wire w postaci zbioru kolejnych szczelin czasowych zapisu „1” bądź „0”. Odczyt danych w sieci MicroLAN, podobnie jak ich zapis, jest sterowany (zarządzany) przez urządzenie master. W celu odczytu danych kontroler ustawia i bada stan magistrali w odpowiedniej szczelinie czasowej, w celu

rozpoznania początku każdego bitu. Odczyt danych z perspektywy mastera jest taki sam, jak szczelina zapisu „1”. Zaczynając od przejścia magistrali ze stanu wysokiego do stanu niskiego (rozpoczęcie każdej kolejnej szczeliny czasowej), urządzenie 1-Wire przesyła jeden bit ze swojego adresu. Jeżeli przesyłany bit jest równy „1”, urządzenie 1-Wire pozostawia wysoki stan magistrali. Jeżeli przesyłane jest „0”, urządzenie 1-Wire zmieni stan magistrali w stan niski (zwarcie linii danych do masy) na czas t_{RDV} równy 15 μs . W tym czasie dane są gotowe do odczytu przez mastera. Czas trwania impulsu startu t_{LOWR} wysyłanego przez kontroler powinien trwać minimalnie 1 μs i nie dłużej niż 15 μs . Po zakończeniu ramki t_{RDV} należy umożliwić powrót napięcia linii 1-Wire do poziomu V_{PULLUP} w czasie $t_{RELEASE}$. Czas trwania impulsu $t_{RELEASE}$ może zmieniać się w zakresie od 0 do 45 μs , przy czym wartość nominalna wynosi 15 μs . Schemat szczeliny czasowej odczytu danych z magistrali 1-Wire przedstawia rysunek 4.5.



Rys. 4.5 Szczelina czasowa odczytu danych z magistrali 1-Wire [11]

Impuls RESET i PRESENCE jest sekwencją rozpoczynającą każdą kolejną operację. Szeregowy rodzaj transmisji i elastyczność sieci 1-Wire, pozwalająca w każdym momencie podłączać/odłączać urządzenia, wymaga kontroli sygnałów przebiegających po magistrali. Do odnotowania przez kontroler, że w sieci znajdują się nowe urządzenie, potrzebny jest odpowiedni sygnał. Swoją „egzystencję” zgłaszają same układy, przy użyciu impulsu obecności (ang. *presence pulse*). Ponadto każdą nową komendę (np. odczyt temperatury z czujnika) inicjowaną przez kontroler poprzedza impuls resetujący stan magistrali (ang. *reset pulse*). Impuls obecności i resetujący są ze sobą ściśle powiązane. Po każdym zrestartowaniu stanu magistrali, układy 1-Wire zgłaszają swoją obecność. Impuls RESET jest pojedynczym sygnałem utrzymującym niski stan magistrali, złożonym z minimum ośmiu kolejnych szczelin czasowych, każda trwająca 480 μs . Po tym magistrala zostaje podniesiona w stan wysoki na czas t_{RSTH} równy 480 μs , czas ten potrzebny jest układom 1-Wire do wysłania sygnału obecności. W czasie t_{RSTH} w sieci MicroLAN nie jest możliwa żadna inna komunikacja. Impuls restartujący zapewnia niezakłócone warunki startu i synchronizację wszystkim następującym szczelinom czasowym. Jeśli układ master wyśle sygnał reset, układy

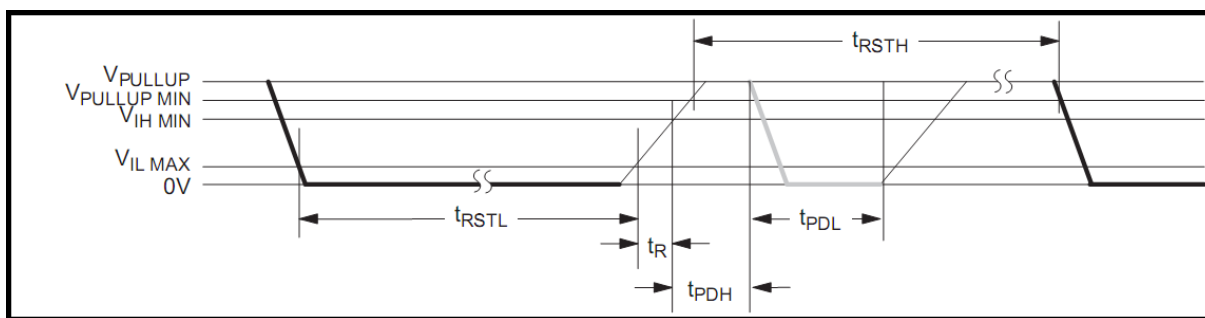
1-Wire odczekają czas t_{PDH} , po czym odpowiedzą sygnałem obecności t_{PDL} ustawiając niski stan magistrali. Dzięki temu kontroler może określić, czy i ile urządzeń jest podłączonych do sieci. Jeżeli w sieci pracuje wiele układów, na podstawie sygnałów obecności, master określić może najgorszy z możliwych warunków pracy sieci (maksymalną długość sygnałów).

Wartości nominalne sygnałów wynoszą 30 μs dla t_{PDH} i 120 μs dla t_{PDL} . Rozważając najgorszy przypadek, zmierzona wartość t_{PDH} określa podstawę czasu najszybszego urządzenia. Suma zmierzonego t_{PDH} i t_{PDL} jest pięć razy większa od podstawy czasu najwolniejszego urządzenia w sieci. Gdy do magistrali podłączone jest tylko jedno urządzenie, obydwie wartości będą odchyłać się w tym samym kierunku. Ta korelacja może być użyta do zbudowania adaptacyjnego systemu. Należy zwrócić szczególną uwagę na kalibrację czasową, po każdym kolejnym resecie, ze względu na zależność charakterystyki czasowej od temperatury pracy i obciążenia poszczególnych układów. Dokładność pomiarów czasy wymaganych poprawnego działania systemu adaptacyjnego jest ograniczony przez charakterystykę logicznego wejścia mastera, stałej czasowej linii 1-Wire (rezystor pull-up x pojemność przewodu) i częstotliwości próbkowania. W przypadku, kiedy czas opadania/narastania zbocza przekracza 1 μs lub gdy najwyższe próbkowanie jest mniejsze niż 1 MHz, nie należy wykorzystywać systemu adaptacyjnego.

Układy 1-Wire zaprojektowane są do pracy w specyficznych warunkach zasilania. Kiedy do układu nie jest dostarczane zasilanie (przez magistralę), może on wykonać żadaną operację przy użyciu ładunku zgromadzonego we wbudowanym kondensatorze (ang. *parasite power*). W zależności od rodzaju operacji, układy 1-Wire muszą być zasilane przez określone minimum czasu, kiedy następuje naładowanie wewnętrznego kondensatora i wymiana danych z układem master. Komendy ROM¹⁹ mogą być przesyłane w sieci MicroLAN niezależnie, jednak operacje pamięciowe muszą być poprzedzone komendą ROM. Zanim układ 1-Wire zaakceptuje wykonanie komendy ROM, w sieci musi odbyć się cykl resetowania i zgłoszenia obecności, trwający minimum 16 szczelin czasowych (0,96 ms). Możliwe jest wprowadzenie urządzeń 1-Wire w stan jałowy, w którym nie będą reagować na sygnały przebiegające po magistrali, do czasu wystąpienia impulsu resetu. W tym czasie możliwa jest tymczasowa komunikacja ze zwiększoną prędkością (ang. *overdrive mode*), o ile stan magistrali nie pozostanie w stanie niskim dłużej niż 120 μs .

Schemat szczeliny czasowej impulsu resetującego i zgłoszenia impulsu obecności przedstawia rysunek 4.6.

¹⁹ ROM ang. Read Only Memory – pamięć tylko do odczytu.



Rys. 4.6 Impuls reset i odpowiadający mu impuls obecności [11]

4.2.3. Unikalny identyfikator

Każdy układ 1-Wire posiada unikalny identyfikator, nadany w czasie procesu produkcji. Jest on podstawowym elementem komunikacji w protokole 1-Wire. Identyfikator stanowi 64-bitowa sekcja ROM, składająca się z:

- sześciobajtowego unikatowego numeru seryjnego,
- ośmiobitowego kodu rodzinnego,
- jednobajtowej sumy CRC.

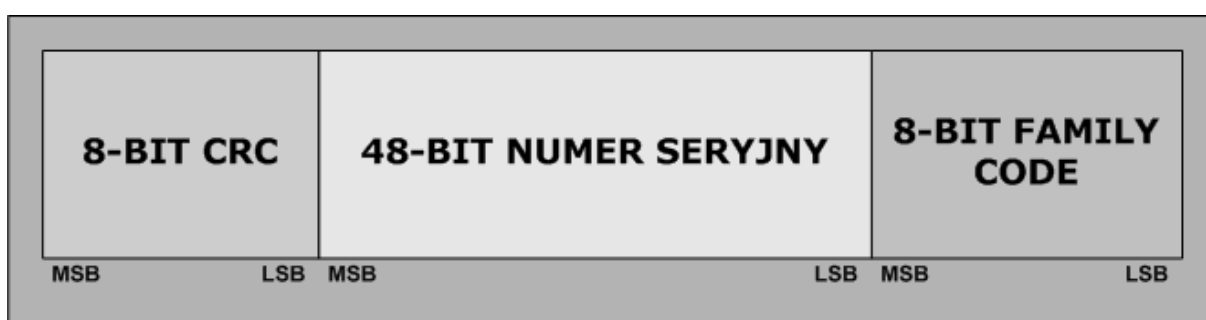
Kod rodzinny (ang. *family code*) jest wartością odnoszącą się do rodzaju i pojemności urządzenia. Siedem mniej znaczących bitów identyfikuje rodzaj urządzenia, najbardziej znaczący bit jest używany do oznaczania serii wyprodukowanych na specjalne zamówienie klientów. Możliwe jest zakodowanie 128 różnych rodzajów (rodzin) urządzeń.

Sześciobajtowy numer seryjny jest reprezentowany przez dowolną liczbę dziesiętną, z przedziału $0 \div 2.81 \cdot 10^{14}$. Jeżeli w ciągu roku wyprodukowanych zostałoby bilion układów posiadających ten sam kod rodzinny, zakres ten wystarczyłby na 281 lat. Ponadto do dyspozycji jest 128 kodów rodzinnych. Kiedy układ posiada ustawiony najbardziej znaczący bit kodu rodzinnego, numer seryjny podlega specjalnym regułom. Dwanaście najbardziej znaczących bitów numeru seryjnego pozwala każdemu z 4096 różnych klientów, na oznaczenie swoich urządzeń. Kod ten przydzielany jest przez producenta układów podczas pierwszego zamówienia. W dyspozycji klienta pozostaje 36 bitów do zagospodarowania na dane wraz z unikatowym numerem seryjnym. Indywidualne układy wymagają specjalnych procedur zamawiania i rejestracji, ponieważ zawierać mogą poufne dane (np. kontrola dostępu do pomieszczeń przy użyciu pastylek iButton).

W zależności od wymagań, klient może wykorzystać wolne 36 bitów na cztery sposoby:

- 8 MSB²⁰ na dane, pozostałe 28 bitów na numer seryjny (268,4 milionów kombinacji),
- 12 MSB na dane, pozostałe 24 bity na numer seryjny (16,8 milionów kombinacji),
- 16 MSB na dane, pozostałe 20 bitów na numer seryjny (1,05 milion kombinacji),
- 20 MSB na dane, pozostałe 16 bitów na numer seryjny (65536 kombinacji).

Identyfikator urządzenia 1-Wire zapisany jest w nieulotnej pamięci ROM. Osiem najmniej znaczących bitów identyfikatora stanowi kod rodzinny, następne 48 bitów to unikalny numer seryjny. Osiem najbardziej znaczących bitów stanowi suma kontrolna CRC, obliczona z pierwszych 56 bitów identyfikatora. Schemat identyfikatora przedstawia rysunek 4.7.



Rys. 4.7 Struktura identyfikatora urządzeń 1-Wire

4.2.4. Obsługa błędów

Najbardziej efektywny schemat wykrywania błędów w komunikacji szeregowej, w której występuje minimalna ilość sprzętu, opiera się o cykliczny kod nadmiarowy (CRC). W czasie, kiedy kontroler rozpoczyna komunikację z urządzeniem 1-Wire, odczytywany jest jego ośmiobajtowy kod ROM, rozpoczynając od LSB²¹. Poprawność komunikacji ustalana jest na podstawie porównania sumy CRC wyliczonej przez kontroler z wartością zapisaną w ostatnim bajcie kodu ROM. W przypadku, kiedy wartości sum kontrolnych nie są równe, występuje błąd i należy powtórzyć komunikację i ponownie odczytać kod ROM. Niektóre układy 1-Wire posiadają pamięć RAM²², do której dostęp może mieć kontroler, przy użyciu odpowiednich komend. Jeśli układ taki nie posiada sprzętowego generatora sum CRC, możliwe jest nadal wykorzystanie sum kontrolnych do weryfikacji operacji zapisu i odczytu pamięci. Do tego celu wykorzystać należy możliwości obliczeniowe kontrolera. W czasie zapisu danych przez kontroler, wyliczona, dołączona i zapisana wraz z danymi zostaje suma CRC. W czasie odczytu sytuacja zostaje odwrócona. Kontroler porównuje wyliczoną sumę z

²⁰ MSB ang. Most Significant Bit – najbardziej znaczący bit.

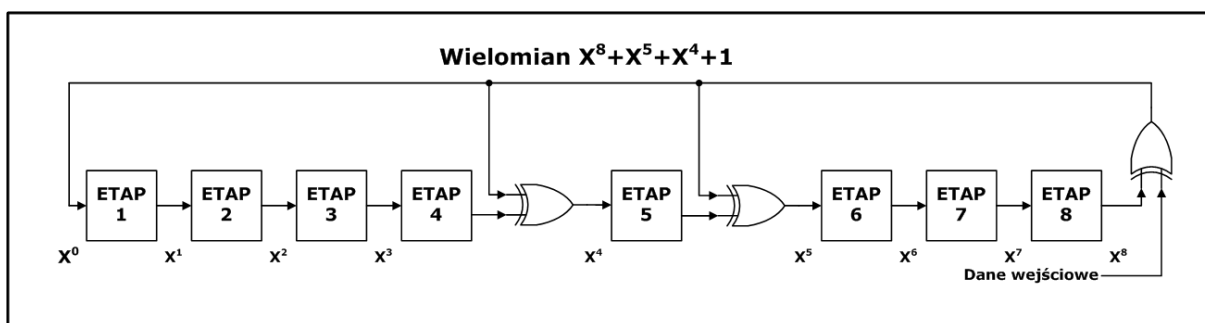
²¹ LSB ang. Least Significant Bit – najmniej znaczący bit.

²² RAM ang. Random Access Memory – pamięć dowolnego dostępu.

wartością zawartą w pamięci układu. Podczas komunikacji w protokole 1-Wire wykorzystuje się ośmiobitową sumę kontrolną. Wykorzystywana jest do sprawdzania kodu ROM i weryfikacji odczytywanych danych. Za pomocą takiego kodu nadmiarowego możliwe jest wykrycie następujących błędów:

- każdej przypadkowej liczby w dowolnym miejscu 64-bitowego kodu,
- wszystkich dwubitowych błędów w dowolnym miejscu 64-bitowego kodu,
- dowolnej sekwencji błędów w ośmiobitowym oknie (1-8 niepoprawnych bitów),
- większości dłuższych sekwencji błędów.

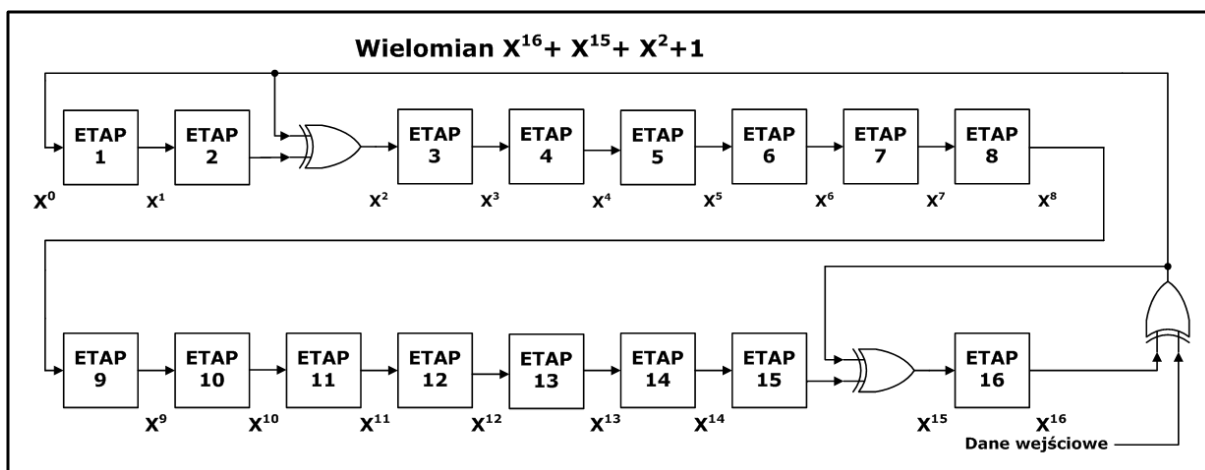
Dane wejściowe są poddawane funkcji różnicy symetrycznej (XOR²³) w rejestrze przesuwным. Rejestr ten może być traktowany, jako matematyczna operacja sekwencyjnego dzielenia. Dane wejściowe są dzielnikiem, a rejestr dzielną. Wyliczony iloraz jest odrzucany, a pozostała reszta jest sumą CRC dla wprowadzonego strumienia danych. Wartość ta pozostaje w rejestrze po ostatnim jego przesunięciu. Rejestr przesuwany i wielomian używany w korekcji błędów transmisji 1-Wire przedstawia rysunek 4.8.



Rys. 4.8 8-bitowy wielomian CRC protokołu 1-Wire [11]

W protokole 1-Wire zapewniono możliwość stosowania bardziej dokładnej 16-bitowej sumy kontrolnej. Wykorzystywana jest ona do przeprowadzania kontroli poprawności danych zapisanych w pamięci RAM, w którą wyposażone są niektóre układy 1-Wire. W operacji korekcji błędów wykorzystywany jest wielomian CRC-16 $X^{16} + X^{15} + X^2 + 1$. Rysunek 4.9 przedstawia schemat sumy kontrolnej stosowanej w protokole 1-Wire.

²³ XOR ang. Exclusive Or – różnica symetryczna, dwuargumentowa funkcja boolowska.



Rys. 4.9 16-to bitowy wielomian CRC protokołu 1-Wire [11]

4.2.5. Korekta pomiarów temperatury

Cyfrowe czujniki temperatury 1-Wire na etapie produkcji kalibrowane są z dokładnością $\pm 0,5^\circ\text{C}$. Dla większości zastosowań dokładność taka jest wystarczająca, jednak w niektórych zastosowaniach, wymagana jest większa precyzja pomiarów. Zastosowanie odpowiedniej techniki pozwala na zwiększenie dokładności cyfrowych czujników temperatury wykonanych przy użyciu technologii półprzewodnikowej. Kompensując przesunięcie i krzywiznę charakterystyki błędów pracy czujnika, może on wykonywać do 10 razy dokładniejsze pomiary. Metoda opiera się o analizę krzywej drugiego stopnia błędów i powtarzalnej natury zależności wyjścia od temperatury.

W celu pokazania powiązania między półprzewodnikową budową czujników a krzywą błędu, praca czujnika temperatury DS1631 została scharakteryzowana w przedziale $-35^\circ\text{C} \div +85^\circ\text{C}$ w „kąpieli wodnej”, a błędy pomiarów w porównaniu do wzorcowego termometru oporowego²⁴ zostały zarejestrowane w odstępach 5°C . Wyniki pomiarów przedstawia rysunek 4.10. Charakterystyka błędów tworzy krzywą drugiego stopnia, którą można opisać równaniem (1):

$$Error = Offset + \alpha \cdot (T_{TS} - T_{ZERO_SLOPE})^2 \quad (1)$$

Gdzie:

T_{TS} jest temperaturą zmierzoną przez czujnik,

α jest współczynnikiem krzywej,

T_{ZERO_SLOPE} jest temperaturą, w której funkcja osiąga minimum,

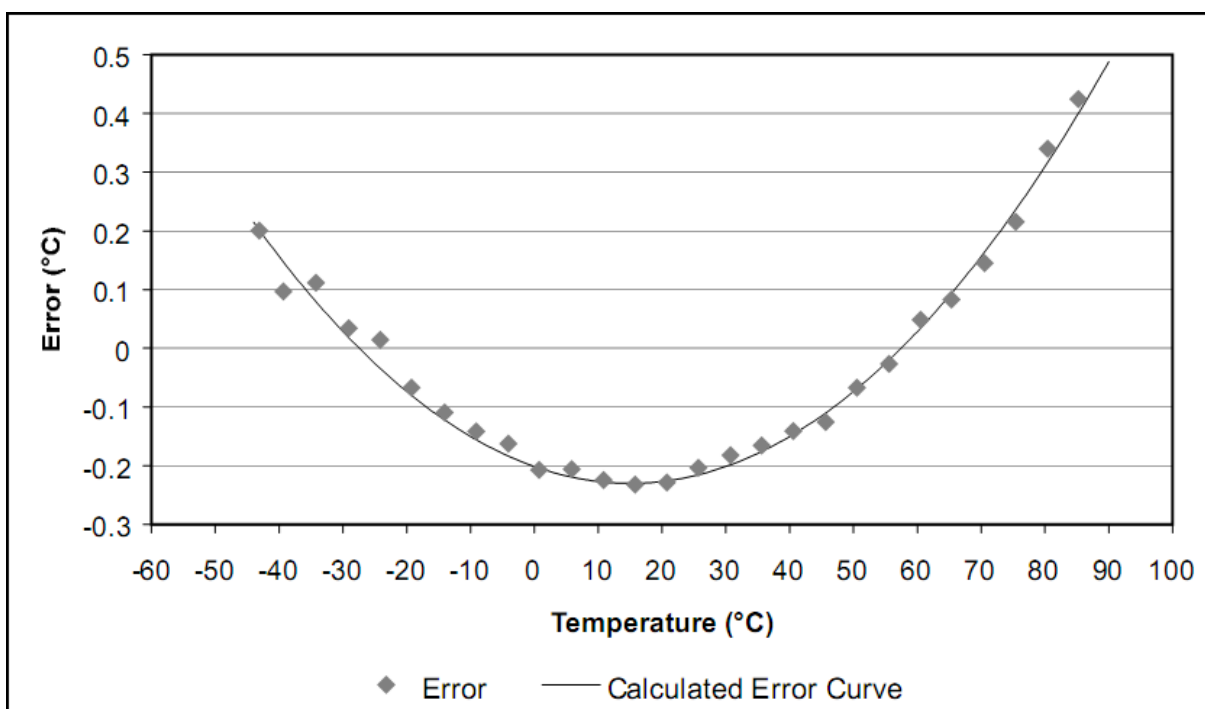
Offset jest wartością błędu w punkcie T_{ZERO_SLOPE} .

²⁴ Termometr oporowy zgodny z normami NIST (ang. National Institute of Standards and Technology)

Po ustaleniu wartości α , $Offset$ i T_{ZERO_SLOPE} w taki sposób, aby dopasować je ściśle do krzywej błędów sensora, możliwe jest wyliczenie przybliżonych wartości błędów dla dowolnej temperatury i skompensowanie pomiaru odejmując wyliczony błąd od wartości zmierzonej. Równanie skompensowanej temperatury przedstawia równanie (2):

$$T_{COMP} = T_{TS} - Error = T_{TS} - [Offset + \alpha \cdot (T_{TS} - T_{ZERO_SLOPE})^2] \quad (2)$$

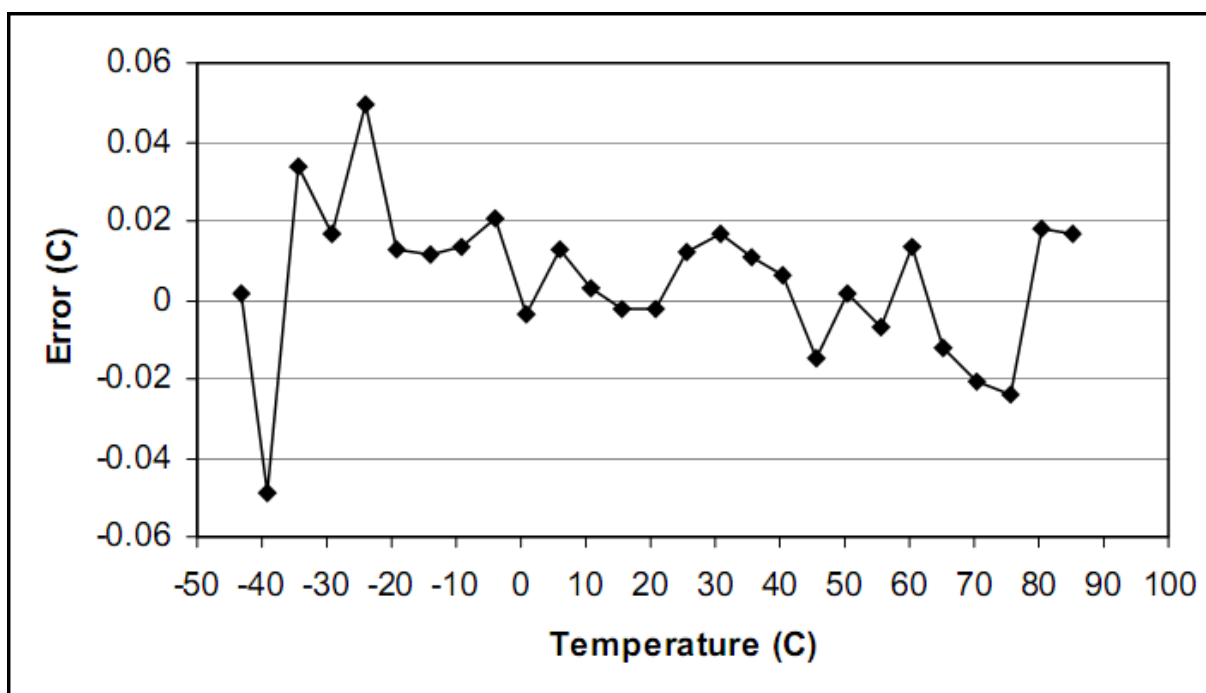
Do osiągnięcia dokładniejszych wyników przy użyciu tej techniki, każdy czujnik powinien być kalibrowany w przedziale temperatury, w którym będzie pracował. Pozwala to na określenie najlepiej dopasowanej krzywej błędów pomiarowych danego sensora.



Rys. 4.10 Wykres wartości błędów pomiarowych i wyliczonej krzywej [10]

W celu zilustrowania techniki kompensacji, zastosowano czujnik DS1631. W tabeli 4.1 przedstawiono zmierzone wartości temperatury i błędy pomiarów w odniesieniu do temperatury otoczenia (zmierzonej przyrządem wzorcowym). Na podstawie równani regresji przedstawionego na rysunku 4.10 można oszacować wartości temperatury T_{ZERO_SLOPE} i $Offset$ odpowiednio 15°C i -0,23°C, dla których $\alpha=1,28 \cdot 10^{-4}$ zapewnia ściśle dopasowanie do krzywej błędów pomiarowych. Podstawiając zmierzone wartości do równania (2) otrzymujemy wartości skompensowanej temperatury, przedstawione w tabeli 4.1. Wartość błędu pomiaru temperatury skompensowanej w odniesieniu do temperatury otoczenia, jest do 10 razy mniejsza od niepewności pierwotnego pomiaru, wykonanego przy użyciu czujnika 1-Wire. Wykres skompensowanej krzywej błędów pomiarowych przedstawia rysunek 4.11.

Należy zaznaczyć, że w celu osiągnięcia najlepszego dopasowania krzywej, pomiar wartości T_{ZERO_SLOPE} , $Offset$ i a trzeba powtórzyć kilkakrotnie.



Rys. 4.11 Wykres krzywej skompensowanych błędów pomiarowych [10]

Tab. 4.1 Wartości kompensowanej temperatury i błędów pomiarowych [10]

Temperatura otoczenia	T_{TS}	Zmierzony błąd	T_{COMP}	Error
-43.1377	-42.9375	0.20015	-43.13603	0.001619
-39.3466	-39.25	0.0966	-39.39572	-0.04912
-34.2368	-34.125	0.11182	-34.20308	0.033737
-29.0969	-29.0625	0.03443	-29.08036	0.016573
-24.1398	-24.125	0.01481	-24.09042	0.049389
-19.2454	-19.3125	-0.06715	-19.23280	0.012547
-14.0779	-14.1875	-0.10964	-14.06626	0.011603
-9.10834	-9.25	-0.14166	-9.09507	0.013266
-4.08784	-4.25	-0.162158	-4.06731	0.020535
0.769446	0.5625	-0.2069459	0.76589	-0.00356
5.831063	5.625	-0.206063	5.84378	0.012717
10.84934	10.625	-0.22434	10.85256	0.003216
15.79473	15.5625	-0.23223	15.79246	-0.00227
20.79082	20.5625	-0.22832	20.78855	-0.00227
25.70361	25.5	-0.20361	25.71593	0.012315
30.74484	30.5625	-0.18234	30.76158	0.016741
35.60317	35.4375	-0.16567	35.61418	0.011006
40.57861	40.4375	-0.14111	40.58489	0.006284
45.68797	45.5625	-0.12547	45.67325	-0.01472
50.56738	50.5	-0.06738	50.56911	0.001733
55.58933	55.5625	-0.02683	55.58245	-0.00688
60.51409	60.5625	0.04841	60.52748	0.01339
65.35461	65.4375	0.08289	65.34273	-0.01188

70.54241	70.6875	0.14509	70.52161	-0.0208
75.40949	75.625	0.21551	75.38579	-0.0237
80.41012	80.75	0.33988	80.42811	0.017987
85.26303	85.6875	0.42447	85.27961	0.016576

4.3. Opis protokołu 1-Wire

4.3.1. Warstwy protokołu

Oprogramowanie lub *firmware*²⁵, które zarządza przesyłaniem danych pomiędzy urządzeniami 1-Wire można odnieść do modelu referencyjnego ISO/OSI. Specyfikacja określa podział protokołu na warstwy: fizyczną, łącza danych, sieci, transportową, sesji, prezentacji i aplikacji. Zgodnie z modelem, wymagania elektryczne i czasowe układów 1-Wire odnoszą się do warstwy fizycznej. Funkcje programowe TouchReset, TouchByte i TouchBit odpowiadają warstwie łącza danych. Funkcje wielopunktowego dostępu First, Next, Access itp., które umożliwiają selekcję poszczególnych węzłów sieci odpowiadają warstwie sieci. Oprogramowanie, które przekazuje dane inne niż zawartość pamięci ROM, pomiędzy poszczególnymi węzłami, odnosi się do warstwy transportowej. W protokole 1-Wire w zależności od środowiska pracy, warstwa sesji może być potrzebna. Warstwa prezentacji zapewnia strukturę plików, która umożliwia urządzeniom 1-Wire wyposażonych w układy pamięciowe, organizację danych w odrębne pliki. Warstwa aplikacji reprezentuje końcową aplikację zaprojektowaną przez użytkownika. Każda warstwa programowa opiera się o zestaw wewnętrznych komend, które w połączeniu pozwalają na wykonanie funkcji (operacji), jaką ma spełniać dany układ 1-Wire (np. pomiar temperatury). Warstwy protokołu 1-Wire przedstawia rysunek 4.12.



Rys. 4.12 Warstwowa architektura protokołu sieciowego 1-Wire

Warstwa fizyczna określa charakterystykę przebiegów elektrycznych, logiczne poziomy napięć i ogólne przedziały czasowe pracy układów 1-Wire.

²⁵ ang. Firmware – oprogramowanie wewnętrzne urządzenia, zapewniające podstawowe procedury i operacje obsługi.

W Warstwie łączy danych zdefiniowana jest podstawowa komunikacja urządzeń 1-Wire. Tutaj też zapewnione są funkcje Reset, Presence (zgłaszanie obecności) i transferu bitów (odczyt i zapis). Po przesłaniu impulsu obecności, komunikacja układów 1-Wire przechodzi do warstwy sieci.

Warstwa sieci umożliwia rozpoznawanie poszczególnych układów, poprzez operacje na identyfikatorze ROM. Ze względu na nazwę identyfikatora, wszystkie operacje jego dotyczące nazywa się funkcjami (komendami) ROM. W tabeli 4.2 zawarto zestawienie komend, kodów odpowiadających poszczególnym komendom i ich krótkiej charakterystyki.

Tab. 4.2 Komendy ROM warstwy sieci protokołu 1-Wire [11]

Nazwa komendy	kod	opis
READ ROM	33H	Identyfikacja urządzenia
		Rozpoznanie czy układy połączone są równolegle.
SKIP ROM	CCH	Pominięcie adresowania, jeśli w sieci pracuje tylko jeden układ.
		„Rozgłoszenie” (ang. <i>broadcast</i>) danych do wszystkich urządzeń w sieci.
MATCH ROM	55H	Adresowanie konkretnego układu w sieci 1-Wire.
SEARCH ROM	F0H	Pobranie identyfikatorów ROM wszystkich układów pracujących w sieci.
		Pobranie numeru seryjnego konkretnego układu i jego ciągłe adresowanie.
OVERDRIVE SKIP ROM	3CH	Ustawienie i rozgłaszanie danych pomiędzy wszystkimi urządzeniami zdolnymi do pracy w trybie overdrive.
OVERDRIVE MATCH ROM	69H	Adresowanie konkretnego układu, ustawienie go w tryb overdrive i kontynuowanie komunikacji w tym trybie.

W celu odczytania identyfikatorów ROM korzystniej jest użyć komendy Search ROM, ponieważ jest ona dostosowana do środowiska wielopunktowego, sieci, w której pracuje więcej niż jeden układ 1-Wire. Po odczytaniu w ten sposób poszczególnych identyfikatorów, możliwe jest sprawdzenie sum kontrolnych CRC, przed rozpoczęciem właściwej komunikacji.

Warstwa transportowa jest odpowiedzialna za transfer danych pomiędzy masterem a segmentami pamięci niezwiązanymi z ROM oraz operacji na obszarze roboczym (ang. *scratchpad*) układów 1-Wire. W tabeli 4.3 zestawiono nazwy i opis komend, wraz z kodami, jakie używane są w tej warstwie protokołu.

Tab. 4.3 Komendy warstwy transportowej protokołu 1-Wire [11]

Nazwa komendy	kod	opis
READ MEMORY	F0H	Odczyt jednego bądź kilku kolejnych bajtów jednej bądź kolejnej strony pamięci o dowolnym poprawnym adresie.
EXTENDED READ MEMORY	A5H	Dotyczy układów z pamięcią EPROM: odczytanie bajtu przekierowania a następnie wykonanie sprawdzenia

		odwróconej sumy CRC16; odczyt kolejnych bajtów o dowolnym poprawnym adresie i wyznaczenie sumy CRC16 w miejscu końca strony.
WRITE SCRATCHPAD	0FH	Dostarczenie adresu celu i zapis od 1 do 32 kolejnych bajtów danych w obszarze roboczym
READ SCRATCHPAD	AAH	Weryfikacja adresu celu i danych zapisanych wcześniej w obszarze roboczym. Odczyt trzech bitów bajtu statusu wskazuje czy wystąpiło przepełnienie obszaru, pojawienie się niekompletnych danych lub czy dane zostały już skopiowane do pamięci.
COPY SCRATCHPAD	55H	Skopiowanie danych przechowywanych w obszarze roboczym pod wskazany adres docelowy. Komenda ta wymaga, aby obszar był wcześniej odczytany. Pozwala to ustalić trzy-bajtowy kod autoryzacyjny, który jest dołączany do komendy Copy Scratchpad.
WRITE MEMORY	0FH	Dotyczy układów z pamięcią EPROM: transfer, weryfikacja i programowanie jednego bądź kolejnych bajtów, rozpoczynając w dowolnym poprawnym adresie obszaru danych pamięci.
WRITE STATUS	55H	Dotyczy układów z pamięcią EPROM: transfer, weryfikacja i programowanie jednego bądź kolejnych bajtów, rozpoczynając w dowolnym poprawnym adresie obszaru statusu pamięci.
READ STATUS	AAH	Dotyczy układów z pamięcią EPROM: odczyt jednego bądź kilku kolejnych bajtów rozpoczynając w dowolnym poprawnym adresie wewnątrz obszaru statusu pamięci. Wyznaczenie odwróconej sumy CRC16 na końcu poszczególnej strony pamięci.

Po przesłaniu i wykonaniu powyższych komend, układy 1-Wire muszą powrócić do warstwy łącza danych w celu wysłania impulsu Reset.

Warstwy łącza danych, sieci i transportowa są fundamentami warstwy prezentacji. Powłoka ta zapewnia zestaw funkcji typowych dla operacji na plikach i katalogach np.: Format, Directory, Type, Copy, Delete, Optimize i sprawdzanie spójności. Dzięki temu układy 1-Wire z wewnętrzną pamięcią mogą być traktowane jak dyskietki.

4.3.2. Medium

Protokół 1-Wire został zaprojektowany, aby zapewnić maksimum możliwości, przy użyciu minimum sprzętu. Do poprawnej pracy urządzeń 1-Wire, potrzebny jest układ master i podłączone do niego układy slave, realizujące różne zadania. Pracą poszczególnych układów jak i samą komunikacją steruje master. Do połączenia układów potrzebne jest medium, które zapewni zasilanie 5 V urządzeniom slave. W zależności od specyfiki danej sieci i odległości układów od kontrolera może to być kabel miedziany (np. skrętka). Układy 1-Wire mogą być elementem urządzenia, zbudowanego na jednej zintegrowanej płycie drukowalnej (wraz z

kontrolerem i/lub z mikroprocesorem). Ponadto w szczególnym przypadku nie jest potrzebne żadne medium, a komunikacja odbywa się poprzez „dotyk”, zetknięcie układu (iButton) z urządzeniem, które steruje komunikacją i jednocześnie zapewnia zasilanie.

4.3.3. Prędkość działania

Protokół 1-Wire dopuszcza komunikację z różną prędkością. Możliwe jest wprowadzenie układów w stan jałowy, kiedy nie reagują na sygnały obecne na magistrali (do wystąpienia kolejnego impulsu reset), w tym czasie możliwa jest komunikacja w trybie Overdrive. W czasie pracy sieci ze zwiększoną prędkością, wszystkie sygnały czasowe (pomijając komendy programowania pamięci EPROM) są dziesięciokrotnie krótsze. Tryb overdrive jest kontrolowany przez układ master. W związku z tym wszystkie układy zdolne do pracy w tym trybie komunikują się z normalną prędkością, o ile szybszy tryb nie zostanie zainicjowany przez kontroler. Układy takie rozpoznać można poprzez ich kod rodzinny lub poprzez użycie specjalnych komend (Overdrive Skip ROM, Overdrive Match ROM), niezrozumiałych dla pozostałych urządzeń. Pierwsze użycie komendy ROM przebiega z normalną (ang. *regular*) prędkością, po czym komunikacja z układami rozpoznanymi i ustawionymi w tryb overdrive przebiega ze zwiększoną prędkością. Dzięki temu, że układy niezdolne do pracy w trybie overdrive pozostają bezczynne, możliwa jest szybsza komunikacja z wieloma urządzeniami w ramach jednej sieci. Impuls reset nominalnej długości, zresetuje wszystkie układy pracujące na magistrali (łącznie z tymi w trybie jałowym) i jednocześnie przywróci urządzenia overdrive do pracy z regularną prędkością. W tabeli 4.4 zestawiono charakterystyki czasowe slotów czasowych przy nominalnej i zwiększonej prędkości.

Tab. 4.4 Porównanie czasu trwania impulsów w trybie regularnym i overdrive

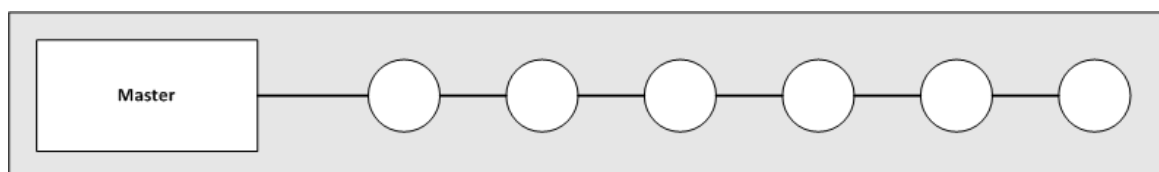
Nazwa slotu czasowego	Tryb regularny	Tryb overdrive
Zapis „1”	$60 \mu\text{s} \leq t_{\text{SLOT}} < 120 \mu\text{s}$	$6 \mu\text{s} \leq t_{\text{SLOT}} < 16 \mu\text{s}$
	$1 \mu\text{s} \leq t_{\text{LOW1}} < 15 \mu\text{s}$	$1 \mu\text{s} \leq t_{\text{LOW1}} < 2 \mu\text{s}$
	$1 \mu\text{s} \leq t_{\text{REC}} < \infty$	$1 \mu\text{s} \leq t_{\text{REC}} < \infty$
Zapis „0”	$60 \mu\text{s} \leq t_{\text{LOW0}} < t_{\text{SLOT}} < 120 \mu\text{s}$	$6 \mu\text{s} \leq t_{\text{LOW0}} < t_{\text{SLOT}} < 16 \mu\text{s}$
	$1 \mu\text{s} \leq t_{\text{REC}} < \infty$	$1 \mu\text{s} \leq t_{\text{REC}} < \infty$
Odczyt	$60 \mu\text{s} \leq t_{\text{SLOT}} < 120 \mu\text{s}$	$6 \mu\text{s} \leq t_{\text{SLOT}} < 16 \mu\text{s}$
	$1 \mu\text{s} \leq t_{\text{LOWR}} < 15 \mu\text{s}$	$1 \mu\text{s} \leq t_{\text{LOWR}} < 2 \mu\text{s}$
	$0 \leq t_{\text{RELEASE}} < 45 \mu\text{s}$	$0 \leq t_{\text{RELEASE}} < 4 \mu\text{s}$
	$1 \mu\text{s} \leq t_{\text{REC}} < \infty$	$1 \mu\text{s} \leq t_{\text{REC}} < \infty$
	$t_{\text{RDV}} = 15 \mu\text{s}$	$t_{\text{RDV}} = 2 \mu\text{s}$
Reset i Presence	$480 \mu\text{s} \leq t_{\text{RSTL}} < \infty$	$48 \mu\text{s} \leq t_{\text{RSTL}} < 80 \mu\text{s}$
	$480 \mu\text{s} \leq t_{\text{RSTH}} < \infty$	$48 \mu\text{s} \leq t_{\text{RSTH}} < \infty$
	$15 \mu\text{s} \leq t_{\text{PDH}} < 60 \mu\text{s}$	$2 \mu\text{s} \leq t_{\text{PDH}} < 6 \mu\text{s}$
	$60 \mu\text{s} \leq t_{\text{PDL}} < 240 \mu\text{s}$	$8 \mu\text{s} \leq t_{\text{PDL}} < 24 \mu\text{s}$

Dokładne określenie prędkości działania sieci 1-Wire jest pojęciem złożonym. Zależy ono od ilości pracujących w niej układów, topologii i odległości układów od mastera, jakości połączenia. Te parametry mają bezpośrednie przełożenie na czas trwania impulsów co wpływa na szybkość komunikacji. Prędkość transmisji w trybie regularnym (ang. *regular*) wynosi 15,4 kbps (sloty czasowe 65 μ s), a w trybie overdrive 100 kbps (sloty 10 μ s). W sieciach „rozległych”, w których układy są znacznie oddalone od kontrolera, możliwe jest przesyłanie danych w trybie elastycznym (ang. *flexible*). W trybie tym sloty czasowe wahają się w zakresie 65 a 72 μ s, co pozwala na komunikację z prędkością 13,9÷15,4 kbps [17].

4.3.4. Topologia sieci 1-Wire

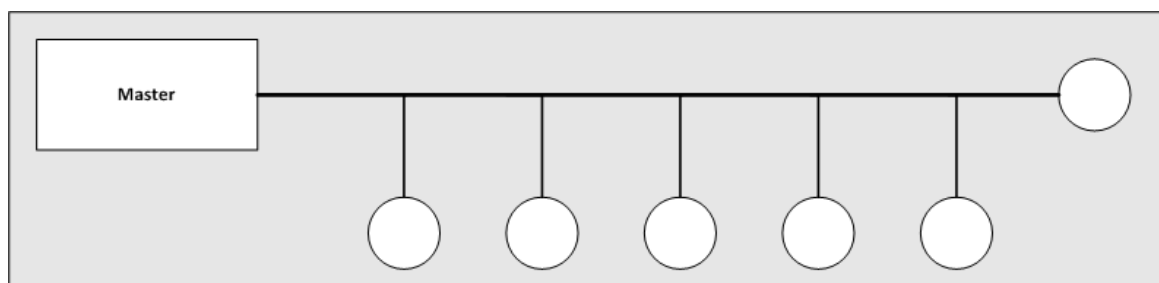
Protokół 1-Wire pozwala na dowolność w budowie połączeń sieciowych, jednak da się je przydzielić do jednej z 3 topologii. Różnią się one między sobą rozmieszczeniem układów wykonawczych i organizacją okablowania. Topologie sieci 1-Wire to [9]:

- Liniowa – Szyna 1-Wire jest prostą parą przewodów od układu master do ostatniego slave’a. Inne układy slave podłączone są równolegle wzdłuż linii, odległość między nimi nie przekracza 3 m.



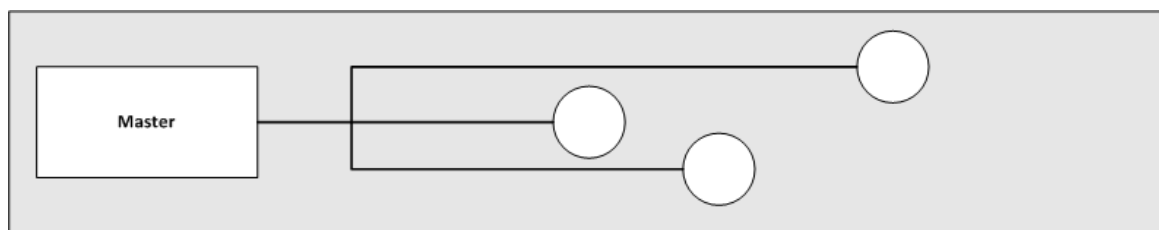
Rys. 4.13 Liniowa topologia sieci 1-Wire [9]

- Odcinkowa (ang. *stubbed*) – Szyna 1-Wire jest pojedynczą linią główną, zaczynając od mastera do ostatniego slave’a. Inne układy slave połączone są z linią główną za pomocą odcinków (odnóg) o długości od 3 m.



Rys. 4.14 Odcinkowa topologia sieci 1-Wire [9]

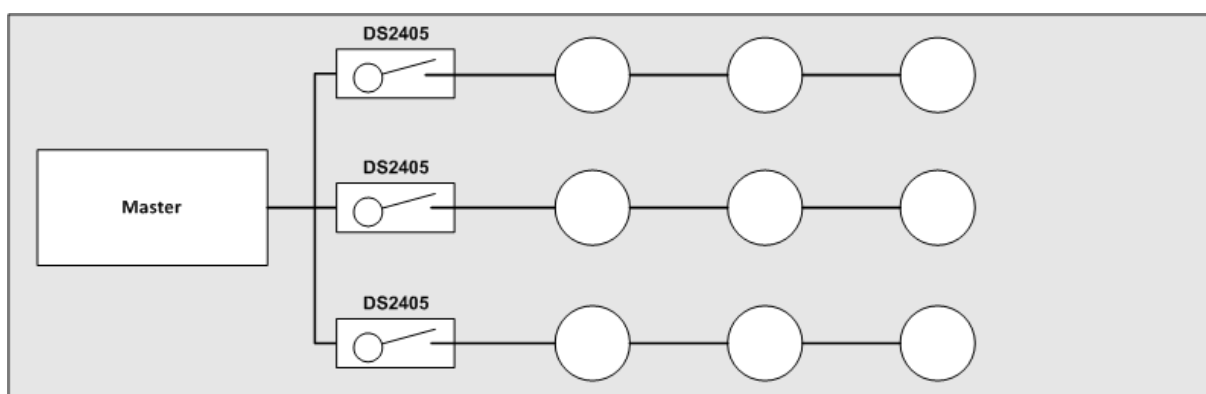
- Gwiazdy – Szyna 1-Wire jest rozszczepiona w bliskiej odległości od mastera i rozdziela się na wiele gałęzi różnej długości. Wzdłuż lub na końcach gałęzi znajdują się układy slave.



Rys. 4.15 Topologia gwiazdy w sieci 1-Wire [9]

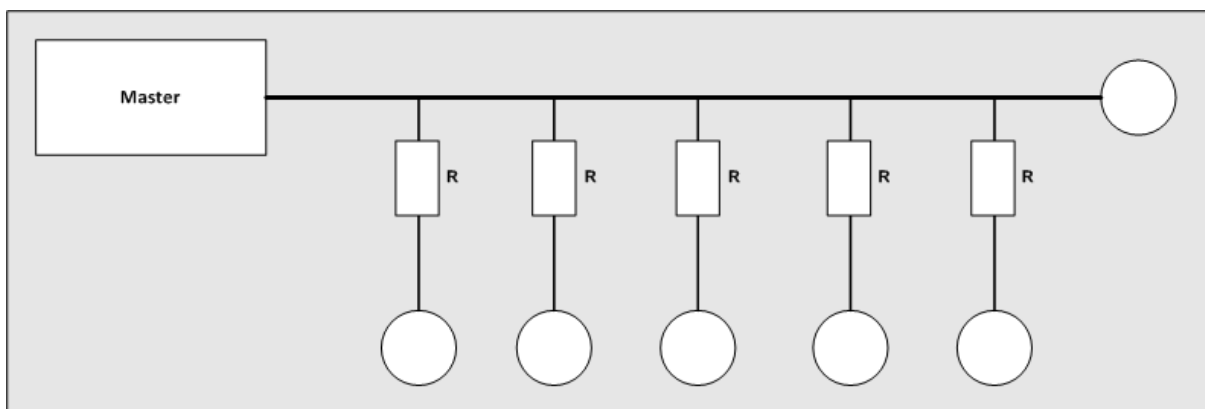
Różna długość poszczególnych gałęzi w topologii gwiazdy powoduje rozbieżność impedancji, jaka dociera do kontrolera. Ponadto odbicia od zakończeń gałęzi mogą powracać w różnym czasie w rezultacie powodując błędy odczytu. Ze względu na powyższe, topologia gwiazdy nie powinna być stosowana. W związku z tym opracowano ulepszone rozwiązanie, które zakłada obecność układów przełączających, co w efekcie pozwala na budowę rozległej sieci.

Dzięki zastosowaniu przełączników (np. DS2409), sieć fizycznie przypomina jedną, a elektrycznie inną topologię. Sieć o topologii gwiazdy, zawierająca przełącznik na początku każdej gałęzi, z perspektywy układu master widziana jest w topologii liniowej (wówczas, gdy jedna gałąź jest włączona).



Rys. 4.16 Topologia przełączana sieci 1-Wire [9]

Kiedy do szyny 1-Wire przyłączony zostaje odcinek, w miejscu połączenia pojawia się różnica impedancji. Odbicia z końca odcinka docierają do trzonu sieci z opóźnieniem zależnym od jego długości. W efekcie w sieci mogą pojawić się sygnały blokujące pracę innych układów slave. Umieszczenie odpowiedniego opornika na danym odcinku pozwala zredukować różnice w impedancji i zmniejszyć amplitudę odbić. Zastosowanie rezystorów $150\ \Omega$ pozwala zmniejszyć niedopasowanie o 20% i zredukować wpływ odbić o 40%. Jednakże dodatkowo wprowadzony opór obniża odporność sieci na zakłócenia o 80%. Użycie rezystorów $100\ \Omega$ pozwala na zachowanie dobrej wydajności, nie obniżając jednocześnie odporności w znaczący sposób.



Rys. 4.17 Topologia rozproszonej impedancji w sieci 1-Wire [9]

4.3.5. Zasięg sieci

W celu omówienia zagadnień związanych z rozmiarem sieci, należy zdefiniować dwa pojęcia, które są podstawą do wyznaczenia zasięgu. Terminy te to: promień i waga. Promień jest to odległość od układu master do najbardziej oddalonego układu slave, wyrażona w metrach. Waga jest miarą długości wszystkich połączonych przewodów w sieci, wyrażoną w metrach. Na przykład sieć w topologii gwiazdy z gałęziami o długości 5 m, 10 m, 15 m ma promień równy 15 m (największa odległość), a wagę 30 m (suma długości). Waga i promień mają bezpośredni wpływ na wydajność sieci. Waga ogranicza czas narastania w przewodach, a promień określa oczekiwanie na najwolniejsze urządzenie w sieci.

Oprócz okablowania, same układy pracujące w sieci mają wpływ na jej rozmiar. Każde z urządzeń 1-Wire zwiększa wagę sieci tak jak krótkie odcinki przewodów, przez co ich oddziaływanie można skwantyfikować. Układy iButton, ze względu na ich budowę, dodają 1 m do wagi sieci, a wszystkie inne układy 0,5 m. W związku z tym, zastosowanie w sieci 100 układów 1-Wire zwiększa jej wagę o 50 m, jednocześnie ogranicza dopuszczalną całkowitą długość okablowania o 50 m. Przy projektowaniu i budowie skomplikowanej sieci 1-Wire należy również wziąć pod uwagę wpływ złączy, płytek montażowych czy zabezpieczeń elektrostatycznych.

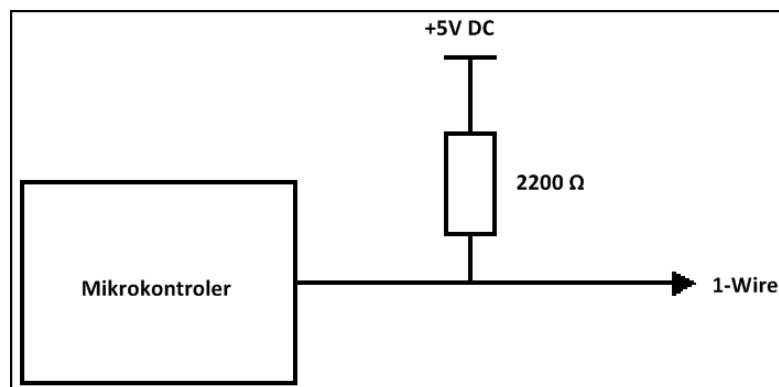
Wpływ na maksymalną wagę i promień sieci ma kilka czynników. Zastosowany w danej sieci interfejs ma wpływ na dopuszczalny rozmiar sieci. Układ master musi zapewnić odpowiedni przepływ prądu, aby przezwyciężyć wagę przewodów i układów slave. Ponadto kontroler dopasowuje długość slotów czasowych do warunków sieci. Impedancja interfejsu powinna być dopasowana do rezystancji sieci, dzięki czemu odbite sygnały nie będą kolidować z pracą pozostałych układów. Promień sieci jest ograniczony przez taktowanie przebiegów odbić, opóźnienie czasowe powodowane przez długość i rezystancję przewodów. Przy promieniu sieci długości 750 m opóźnienie sygnału docierającego od najdalszego układu

do mastera przekracza możliwości protokołu. Waga sieci jest ograniczona przez zdolność okablowania do szybkiego ładowania i rozładowywania. Prosty rezystor podwyższający jest ograniczony do około 200 m. Rozbudowane interfejsy, aktywnie podwyższające prąd w sieci, pozwalają obsłużyć wagę do 500 m.

Kontroler sieci 1-Wire oprócz zadań związanych z komunikacją, jest odpowiedzialny za zasilanie układów pracujących w trybie „pasożytniczym” (pobierających zasilanie z linii danych). Każde takie urządzenie kradnie zasilanie z magistrali, kiedy napięcie szyny jest wyższe od napięcia wewnętrznego kondensatora. Kiedy waga sieci wzrasta, mogą wystąpić przypadki, kiedy prąd dostarczany przez kontroler, jest niewystarczający do zapewnienia zasilania układom slave. Najbardziej niekorzystnym dla „pasożytniczych” układów są sytuacje, w których master przesyła długie serie zerowych bitów. Ze względu na utrzymywanie magistrali w niskim stanie, może zabraknąć okazji do naładowania kondensatora. Napięcie wewnątrz układów slave maleje, ich zdolność do przesyłania zer spada, przez co wydłuża się taktowanie slotów czasowych. Wówczas napięcie może osiągnąć poziom krytyczny, kiedy urządzenie slave wejdzie w tryb reset i przestanie odpowiadać. Ponowny wzrost napięcia na magistrali spowoduje naładowanie „uśpionego” układu, co w rezultacie może doprowadzić do ingerencji w komunikację innych układów pracujących w sieci. W sytuacji, gdy sieć ma niewystarczającą energię do zasilenia urządzeń slave, awarie związane z przesyłaniem danych będą się sporadycznie pojawiać.

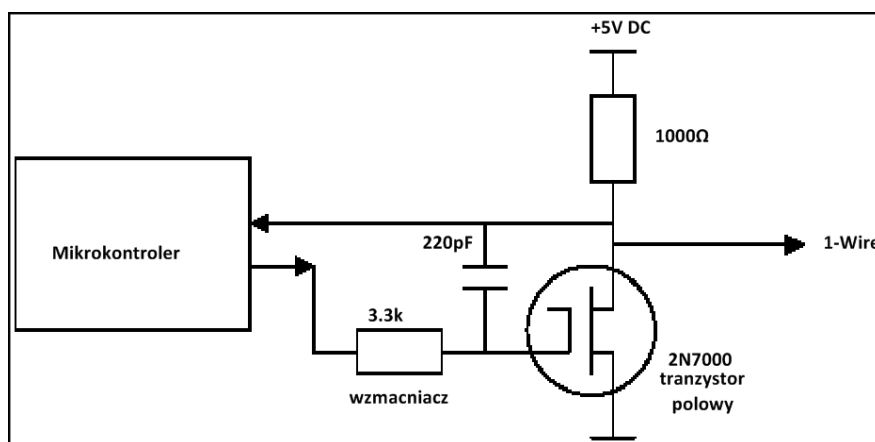
Na potrzeby różnych sieci opracowano różne interfejsy. Każdy z nich dostosowany jest do sieci o różnym zasięgu (mały, średni, duży). Ponadto kontroler dużego zasięgu małej sieci nie będzie tak wydajny, jak ten do niej wyspecjalizowany. Rodzaje interfejsów i zalecane ograniczenia opisano poniżej [9].

Proste mikrokontrolery wyposażone jedynie w wyjście, nie są w stanie zapewnić odpowiedniego prądu dla dużej czy średniej sieci. Zasięg sieci ogranicza się maksymalnie do 3 m długości wagi i promienia.



Rys. 4.18 Prosty interfejs sieciowy [9]

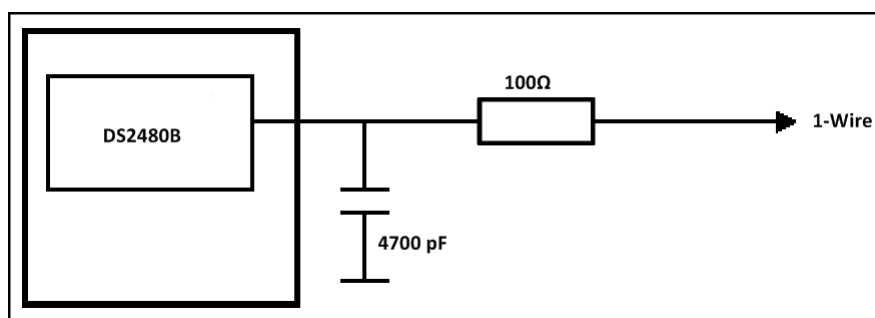
Mikrokontrolery zawierające tranzystory polowe z kontrolą szybkości narastania i rezystorem podwyższającym $1\text{ k}\Omega$ obsługują sieci średniego zasięgu. Interfejs obsługuje poprawnie sieci o wadze i zasięgu do 200 m.



Rys. 4.19 Interfejs wyposażony w tranzystor i wzmacniacz [9]

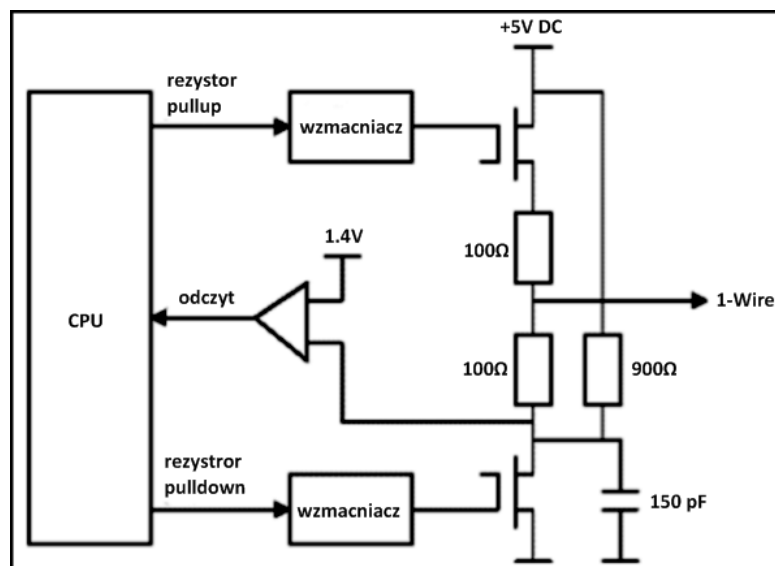
Port równoległy DS1410E i szeregowy port DS90C07 zostały zaprojektowane do użycia z bardzo krótkimi (w zasięgu kontrolera) sondami iButton. Nie nadaje się do zastosowania w sieciach 1-Wire średniego i długiego zasięgu. Promień nie powinien przekraczać długości 3 m, a waga 40 m.

Interfejsy zbudowane z wykorzystaniem układu DS2480B, stosowane w sieciach dłuższych niż 1 m, powinny być dodatkowo wyposażone w filtr RC. Dzięki obecności filtra, układ master może obsługiwać sieci o promieniu do 200 m. Układ DS2480B działa z wykorzystaniem zmiennego taktowania, które pozwala na zwiększenie niezawodności i wydajności sieci. Taktowanie może być sterowane przez oprogramowanie, dzięki czemu da się je poprawnie dopasować do warunków danej sieci.



Rys. 4.20 Interfejs sieciowy z filtrem RC [9]

Zaawansowany interfejs jest układem zaprojektowanym specjalnie dla sieci długiego i małego zasięgu. Dzięki skomplikowanej budowie pozwala na komunikację w sieci o wadze i promieniu długości do 500 m.



Rys. 4.21 Zaawansowany interfejs sieciowy [9]

5. Układy pracujące w technologii 1-Wire

5.1. Wprowadzenie

Specyfika pracy, prostota budowy i łatwość implementacji umożliwia stosowanie produktów 1-Wire w szerokiej gamie zastosowań. Układy można podzielić na kilka kategorii: układy pamięciowe, czujniki temperatury, interfejsy, przetworniki A/C, zegary, układy monitorujące pracę baterii (akumulatorów). Odrębną grupę stanowią urządzenia serii *iButton*, są to miniaturowe banki danych, wykorzystywane głównie do kontroli dostępu (autoryzacji). Urządzenie przypominające baterię zegarkową, które posiada własną pamięć (do 64 kb), unikalny identyfikator i niewymagające do pracy własnego zasilania, może znaleźć wiele zastosowań.

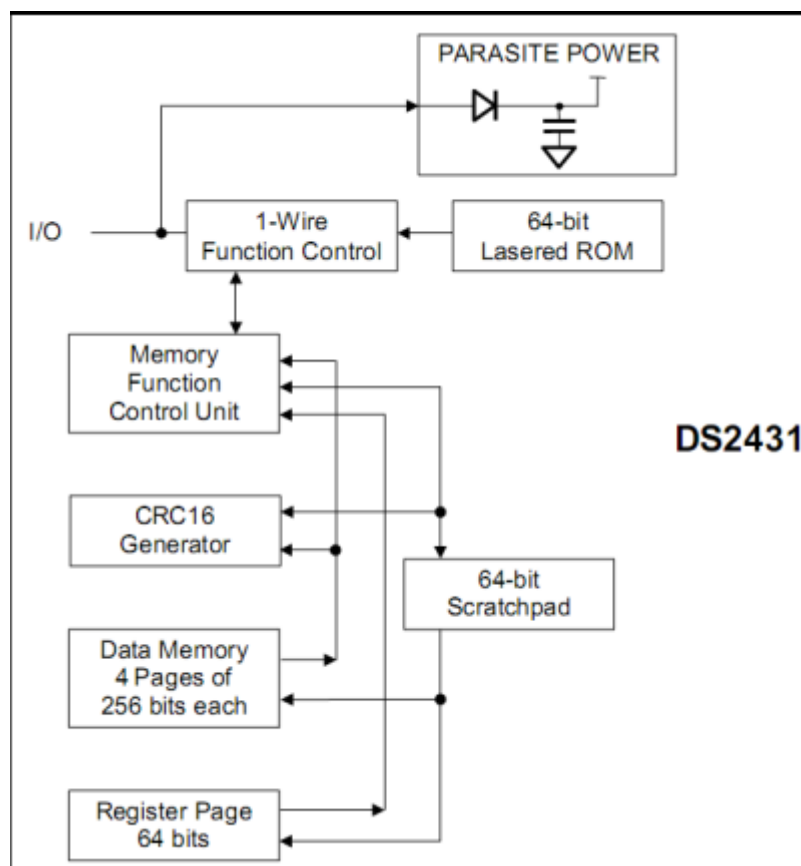
Wszystkie układy 1-Wire mogą być wykorzystywane do różnych funkcji, np.:

- Identyfikacja „elektroniczne etykiety” – kartridże do drukarek, glukometry i paski, płytki PCB, moduły RACK, baterie i akcesoria do telefonów komórkowych;
- Zabezpieczenia sprzętowe – tunery telewizji, zasilacze do laptopów;
- Monitory pracy – kalibracja urządzeń medycznych, kalibracja pracy/ladowania akumulatorów, monitorowanie temperatury/wilgotności/zawartości CO₂;
- Przechowywanie danych – cała dokumentacja medyczna pacjenta na jednej „pastylce” *iButton*.

5.2. Opis układów 1-Wire

5.2.1. Układy pamięciowe

Układy pamięciowe 1-Wire służą do zapisywania danych przez użytkownika. Rozróżnić można 2 rodzaje układów, z pamięcią EPROM²⁶ i EEPROM²⁷. Poszczególne układy zorganizowane są w kilka stron pamięci ośmiobajtowej (w zależności od pojemności). Jak każde urządzenie przystosowane do pracy w sieci 1-Wire, pamięci posiadają unikalny identyfikator ROM (64 bity). Zapis danych do pamięci odbywa się za pośrednictwem obszaru roboczego (ang. *scratchpad*). W pierwszej kolejności żądane informacje umieszczane są w tym obszarze, następnie układ master odczytuje te dane w celu weryfikacji poprawności. Informacje weryfikowane są sumą kontrolną CRC-16. Po potwierdzeniu spójności danych, układ master wydaje polecenie przekopiowania zawartości obszaru roboczego, do ustalonego miejsca w pamięci układu. Odczyt i zapis danych do pamięci rozpoczyna się zawsze od LSB. Diagram blokowy układu DS2431 przedstawia rysunek 5.1.



Rys. 5.1 Diagram blokowy układu DS2431 - 1Kb EEPROM [15]

²⁶ EPROM (ang. Erasable Programmable Read-Only Memory) – rodzaj nieulotnej pamięci elektrycznej

²⁷ EEPROM (ang. Electrically-Erasable Programmable Read-Only Memory) – nieulotna pamięć programowana elektrycznie. W przeciwieństwie do EPROM, można ją kasować elektrycznie.

W tabeli 5.1 zestawiono układy pamięciowe 1-Wire wraz z krótkim opisem cech charakterystycznych.

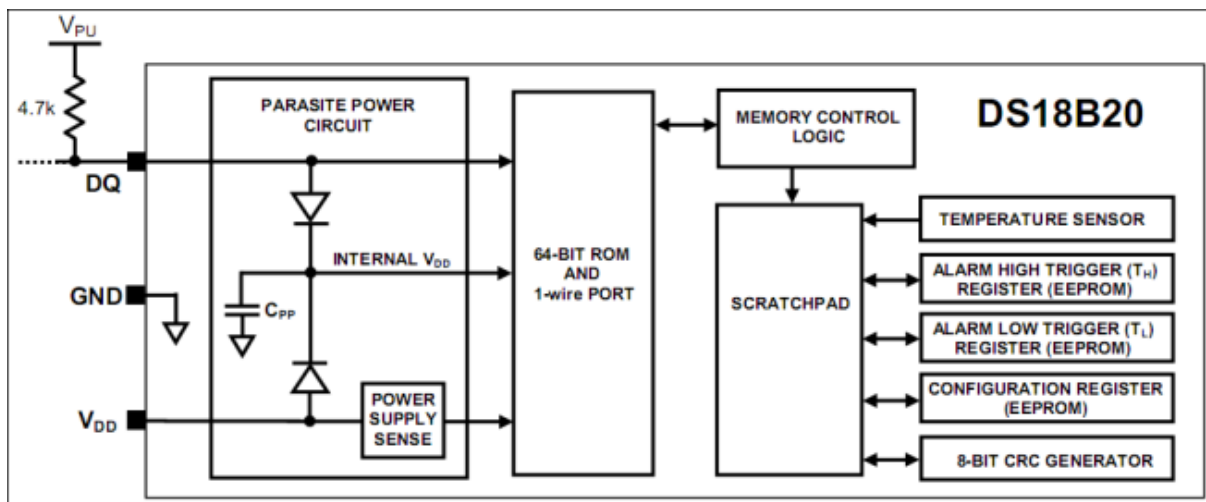
Tab. 5.1 Układy pamięciowe 1-Wire

Symbol	Nazwa	Opis
DS2401	1-Wire Silicon Serial Number	Tylko identyfikator ROM
DS2411	1-Wire Vcc Powered Silicon Serial Number	Tylko identyfikator ROM, zakres napięcia pracy 1,5 V÷5,25 V wymaga zewnętrznego zasilania
DS2431	1024-Bit 1-Wire EEPROM	Pamięć zapis-odczyt, 4 strony po 8 bajtów
DS2432	1kbit SHA-1 Protected EEPROM	Pamięć, której zapis i odczyt chroniony jest algorytmem SHA-1
DS2433	4 kbit 1-Wire EEPROM	16 stron po 8 bajtów
DS2502	1-Wire 1k-bit EPROM	Pamięć tylko do dopisywania
DS2502-E64	1-Wire IEEE EUI-64 Node Address Chip	Odmiana DS2502 z globalnym adresem węzła IEEE 1394-1995 (FireWire).
DS2502-UNW	1-Wire 1k-bit UniqueWare EPROM	Odmiana DS2502 z identyfikatorem UniqueWare
DS2505	1-Wire 16k-bit EPROM	Pamięć tylko do dopisywania, 64 strony pamięci
DS2505-UNW	1-Wire 16k-bit UniqueWare EPROM	Odmiana DS2505 z identyfikatorem UniqueWare
DS2506	1-Wire 64 kbit EPROM	Pamięć tylko do dopisywania, 256 stron
DS2506-UNW	1-Wire 64k-bit UniqueWare EPROM	Odmiana DS2506 z identyfikatorem UniqueWare
DS28E01-100	1K-Bit Protected 1-Wire EEPROM with SHA-1 Engine	Odmiana DS2432 z pinem IO
DS28E04-100	4096-Bit Addressable 1-Wire EEPROM with PIO	Odmiana DS2433 z 7-bitowym wejściem połączonym z adresem i 2 portami IO

5.2.2. Czujniki temperatury

Seria czujników temperatury, pracujących w technologii 1-Wire, charakteryzuje się 9-cio bitową rozdzielczością pomiaru i możliwością ustalenia wartości alarmowych. Zakres pomiarowy układów wynosi $-55^{\circ}\text{C}\div 125^{\circ}\text{C}$ z dokładnością od $\pm 0,5^{\circ}\text{C}$ do $\pm 2^{\circ}\text{C}$ (w przedziale $-10^{\circ}\text{C}\div 85^{\circ}\text{C}$). Maksymalny czas konwersji pomiaru temperatury wynosi 750 ms. Pomiar przechowywany jest w 2-bajtowym rejestrze obszaru roboczego, a zdefiniowane wartości

alarmowe w nieulotnej pamięci EEPROM. Dane pomiarowe sprawdzane są przy pomocy sumy kontrolnej CRC-8. W zależności od wersji, niektóre czujniki temperatury mogą pracować z większą, regulowaną rozdzielczością (do 12 bitów, tab. 5.2). Schemat blokowy czujnika DS18B20 przedstawia rysunek 5.2.



Rys. 5.2 Czujnik temperatury 1-Wire DS18B20 [12]

Tabela 5.2 zawiera zestawienie modeli czujników temperatury 1-Wire wraz z opisem cech charakterystycznych.

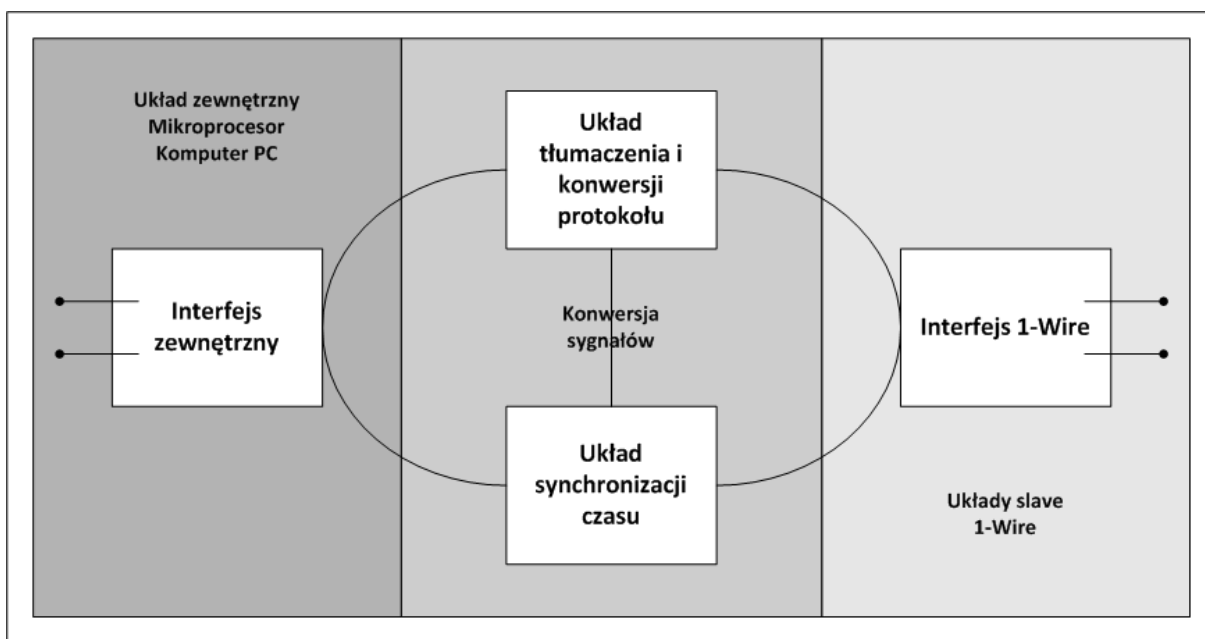
Tab. 5.2 Czujniki temperatury 1-Wire

Symbol	Nazwa	Opis
DS1821	Digital Thermometer or Standalone Thermostat	Czujnik rozdzielczości 8-bitów, dokładność $\pm 1^{\circ}\text{C}$ w przedziale $0 \div 85^{\circ}\text{C}$. W funkcji termostatu na wyjściu DQ pokazuje stan, w zależności od ustalonych alarmów.
DS1822	Econo 1-Wire Digital Thermometer	Rozdzielczość regulowana $9 \div 12$ bitów. Dokładność $\pm 2^{\circ}\text{C}$ w przedziale $-10 \div 85^{\circ}\text{C}$
DS1822PAR	Econo Parasite-Power Digital Thermometer	Odmiana DS1822 pracująca tylko przy zasilaniu z linii zasilania („pasożytniczym”). Zakres pracy $-55 \div 100^{\circ}\text{C}$
DS18B20	High-Precision 1-Wire Digital Thermometer	Regulowana rozdzielczość $9 \div 12$ bitów. Dokładność $\pm 0,5^{\circ}\text{C}$ w przedziale $-10 \div 85^{\circ}\text{C}$.
DS1825	Programmable 1-Wire Digital Thermometer with 4-bit Address	Odmiana DS18B20. Cztery bitowy adres ustalany za pomocą pinów.
DS18B20PAR	High-Precision Parasite-Power Digital Thermometer	Odmiana DS18B20 przeznaczona do pracy tylko z zasilaniem „pasożytniczym”.

DS18S20	1-Wire Digital Thermometer	Rozdzielczość 9-bitowa, dokładność $\pm 0,5^{\circ}\text{C}$
DS18S20PAR	Parasite-Power Digital Thermometer	Odmiana DS18S20 tylko z zasilaniem „pasożytniczym”
DS28EA00	1-Wire Digital Thermometer with Sequence Detect and PIO	Regulowana rozdzielczość $9\div 12$ bitów. Dokładność $\pm 0,5^{\circ}\text{C}$, zakres pracy $-40\div 85^{\circ}\text{C}$

5.2.3. Interfejsy Komunikacyjne

Specyfikacja protokołu 1-Wire określa, że transmisja, sterowanie, odczyt i zapis danych odbywa się za pomocą układu master. Pracą takiego układu steruje natomiast komputer PC lub mikroprocesor, który realizuje określony przez użytkownika program. Interfejs 1-Wire tłumaczy sygnały specyficzne dla tego protokołu, na „język” sygnałów protokołu docelowego. W celu dopasowania dwóch różnych sygnałów, sterownik wyposażony jest w układ przetwarzania sygnałów w czasie. W sterownikach 1-Wire występują też układy dopasowujące poziomy napięcie. Schemat ideowy interfejsu 1-Wire przedstawia rysunek 5.3.



Rys. 5.3 Schemat ideowy interfejsu 1-Wire

W tabeli 5.3 przedstawiono interfejsy przeznaczone do komunikacji z układami 1-Wire.

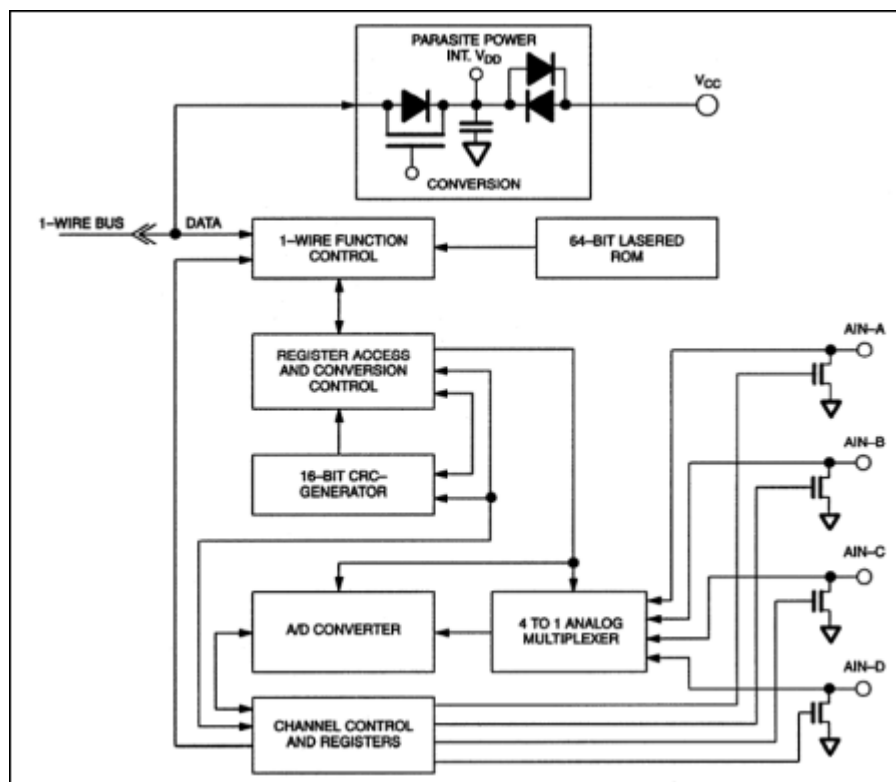
Tab. 5.3 Spis interfejsów 1-Wire wraz z opisem

Symbol	Nazwa	Opis
DS1482	1-Wire Level Shifter and Line Driver with Load Sensor	Przejęcie z poziomu 3 V na 5 V. Możliwość podłączenia bezpośrednio do synchronicznego portu szeregowego, jeśli prędkość transmisji odpowiada taktowaniu 1-Wire.
DS1WM	Synthesizable 1-Wire Bus Master	Interfejs sterowany przez CPU. Procesor podaje polecenie, odpowiednie taktowanie zapewnia układ. W odpowiedzi procesor

		otrzymuje przerwanie.
DS2480B	Serial 1-Wire TM Line Driver with Load Sensor	Uniwersalny sterownik portu szeregowego. Możliwa praca z 4 różnymi prędkościami 115,2 kbps, 57,6 kbps, 19,2 kbps i domyślnym 9,6 kbps. Generator czasu kontroluje wszystkie zależności czasowe interfejsu, jak i przebiegi czasowe protokołu 1-Wire.
DS2482-100	Single Channel 1-Wire Master	Bezpośrednia konwersja do protokołu I ² C w trybie standardowym (100 kHz) i szybkim (400 kHz). Wyposażony w 2 wejścia binarne do przydzielenia adresu, w celu uniknięcia konfliktów z innymi układami I ² C.
DS2482-800	8 Channel 1-Wire Master	Odmiana układu DS2482-100 wyposażony w 3-bitowy adres i 8 niezależnych kanałów 1-Wire.
DS2490	USB to 1-Wire Bridge Chip	Bezpośredni interfejs między protokołem 1-Wire a magistralą szeregową USB. Układ umożliwia zdalne wzbudzenie „uśpionego” komputera, do którego podłączony jest sterownik.

5.2.4. Przetworniki A/C

W celu konwersji sygnałów do postaci cyfrowej z układów analogowych, pracujących w danej sieci, należy zastosować przetwornik A/C. Układ ten pozwala zmierzyć napięcie na wejściu a następnie przekazać je do układu master w postaci cyfrowej. Układ DS2450 stosuje konwersję sukcesywnej aproksymacji (ang. *successive-approximation*) do mierzenia napięcia wejściowego. Posiada cztery odrębne kanały, każdy wyposażony we własny rejestr do przechowywania wejściowego poziomu napięcia, rozdzielczości, poziomów alarmowych jak i zestawu flag pomocnych przy wyszukiwaniu warunkowym. Dwie flagi alarmowe dla każdego kanału wskazują czy zmierzone napięcie jest zbyt wysokie lub niskie, bez potrzeby wykonywania porównań przez układ master. Każda konwersja A/C jest inicjowana przez sterownik sieci. Nieużywany kanał może służyć, jako cyfrowe wyjście z otwartym drenem (ang. *open-drain*). Po wyłączeniu wejścia, układ master może bezpośrednio włączać lub wyłączać tranzystor z otwartym drenem, w wybranym kanale. Schemat blokowy przetwornika DS2450 przedstawia rysunek 5.4.



Rys. 5.4 Schemat blokowy układu DS2450 [16]

5.2.5. Układy przetwarzania czasu

W niektórych sytuacjach konieczne jest, aby pewnie zdarzenia odbywały się cyklicznie, w określonych odstępach czasu. W tym celu producent układów 1-Wire, opracował układy przetwarzania czasu. Są to zegary czasu rzeczywistego RTC²⁸ zbudowane na podstawie binarnego licznika (rozdzielczość 32-bitowa). Za pomocą tych układów można odmierzać czas z odstępem 1 sekundy, a rozdzielczość licznika pozwala na zliczenie 136 lat. Czas/data reprezentowana jest przez ilość sekund, jaka upłynęła od stempla czasowego, zdefiniowanego przez użytkownika. Układ DS2417 i DS2422 pozwalają na wysyłanie cyklicznego przerwania, w odstępie od 1 sekundy do 36 godzin (273 godzin w przypadku układu DS2422). Taktowanie zegara odbywa się przy użyciu oscylatora kwarcowego o częstotliwości 32,768 kHz. Układy przetwarzania czasu pracujące w sieci 1-Wire odznaczają się dokładnością ± 2 minut w przeciągu miesiąca w temperaturze 25°C, przy czym zakres pracy układów wynosi -40°C÷85°C. W tabeli 5.4 zestawiono układy przetwarzania czasu wraz z opisem.

Tab. 5.4 Układy RTC pracujące w sieci 1-Wire

Symbol	Nazwa	Opis
DS2415	1-Wire Time Chip	Podstawowy zegar czasu rzeczywistego
DS2417	1-Wire Time Chip with Interrupt	Odmiana układu DS2415 z generatorem

²⁸ RTC (Real Time Clock) – zegar czasu rzeczywistego

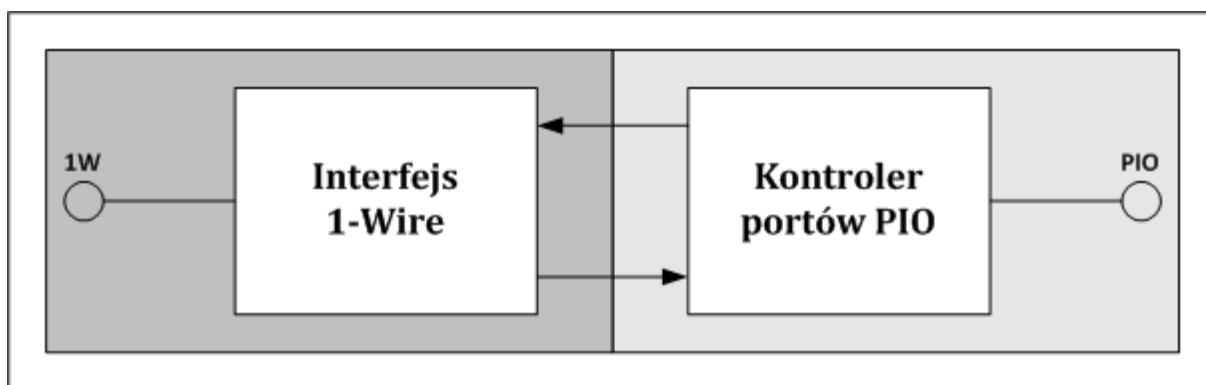
		przerwań
DS2422	1-Wire Temperature/Datalogger with 8kB Datalog Memory	Czujnik temperatury, zegar RTC, pamięć, interfejs 1-Wire. Dodatkowe wyjścia na zewnętrzny przetwornik A/C i źródło prądowe.

5.2.6. Przełączniki

Wraz ze wzrostem ilości układów w sieci MicroLAN, maleje możliwość dalszej rozbudowy. W celu zwiększenia rozmiarów sieci należy zastosować jeden bądź więcej układów przełączających. Dzięki zastosowaniu przełączników, możliwe jest podzielenie sieci na segmenty, które w razie potrzeby, będą włączane lub wyłączane. Pracą układów przełączających, jak ich wszystkich innych podłączonych do sieci 1-Wire steruje układ master. Zastosowanie przełączników nie ogranicza się tylko do segmentacji sieci, możliwe jest dzięki nim sterowanie innymi układami, np. [14]:

- układ slave dla mikroprocesora,
- wielozadaniowy czujnik/aktor,
- interfejs inteligentnego wyświetlacza,
- zwiększenie ilości portów mikrokontrolera,
- skaner klawiatury sterowany mikroprocesorem,
- czujnik wciśnięcia przycisków sterowany „pasożytniczo”.

Układy przełączające 1-Wire zbudowane są jednego lub więcej wyjść w postaci tranzystorowego portu wejścia/wyjścia z otwartym drenem. Schemat ideowy przełącznika 1-Wire przedstawia rysunek 5.5.



Rys. 5.5 Przełącznik 1-Wire – schemat ideowy

Poszczególne porty PIO w przełącznikach mogą być włączane/wyłączane, a ich stan zapisywany jest w odpowiednich rejestrach. Dzięki temu stan portów może być odczytywany przez układ master. Przy użyciu funkcji 1-Wire wyszukiwania warunkowego (ang. *Conditional Search ROM [ECh]*) możliwe jest sterowanie układami, które spełniają określone

kryteria, np. znajdź wszystkie aktywne układy. Domyślnym stanem portu po uruchomieniu układu jest stan wyłączony. W tabeli 5.5 przedstawiono spis układów przełączających 1-Wire wraz z opisem.

Tab. 5.5 Przełączniki 1-Wire

Symbol	Nazwa	Opis
DS2405	1-Wire Addressable Switch	1 port PIO, maksymalne napięcie 5 V
DS2406	1-Wire Dual Addressable Switch Plus 1kbit Memory	2 porty PIO, maksymalne napięcie PIO-A 13 V, PIO-B 6,5 V; dodatkowo 1 kb pamięci EPROM
DS2409	1-Wire 8-channel Port Expander	8 portów PIO, maksymalne napięcie 6 V
DS2413	1-Wire Dual Channel Addressable Switch	2 porty PIO, maksymalne napięcie 28 V

5.2.7. Pomiar parametrów mikroklimatu z wykorzystaniem układów 1-Wire

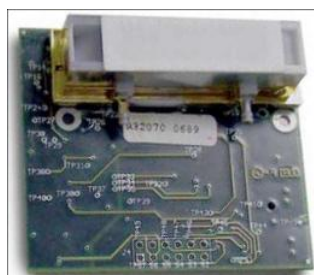
Pomiar wielkości takich jak wilgotność powietrza i stężenia CO₂ w sieci 1-Wire możliwy jest przy wykorzystaniu przetwornika A/C DS2450 i odpowiednich przetworników. Producent układów 1-Wire nie posiada w swojej ofercie układów pomiarowych przystosowanych bezpośrednio do pracy w sieci MicroLAN (poza grupą czujników temperatury), z wyjątkiem pastylki iButton DS1923.

W celu pomiaru stężenia CO₂ użyć można czujnika Telaire 6004. Sensor wykorzystuje metodę absorpcji promieniowania podczerwonego (ang. *non dispersive infrared*). Zaletą tego czujnika jest możliwość pracy bez konieczności wymuszonego przepływu powietrza. Parametry układu zestawiono w tabeli 5.6.

Tab. 5.6 Parametry układu Telaire 6004 [29]

Parametr	Wartość
Zakres pomiarowy	0÷2000 ppm
Dokładność pomiaru	±40 ppm względem kalibracji fabrycznej
Czas odpowiedzi	<60 s dla zmiany kroku 90%
Czas próbkowania	1,6 s
Gotowość do pomiaru	<2 min (działanie), 10 min (maks. dokładność)
Warunki pracy	0÷50°C
Sygnał wyjściowy	Cyfrowy UART@9600 bodów, Analogowy 0÷4 V DC=0÷2000 ppm
Zasilanie	5 V DC
Pobór prądu	średni 45 mA, szczytowy 135 mA

Na rysunku 5.6 przedstawiono moduł pomiaru ciśnienia 6004.

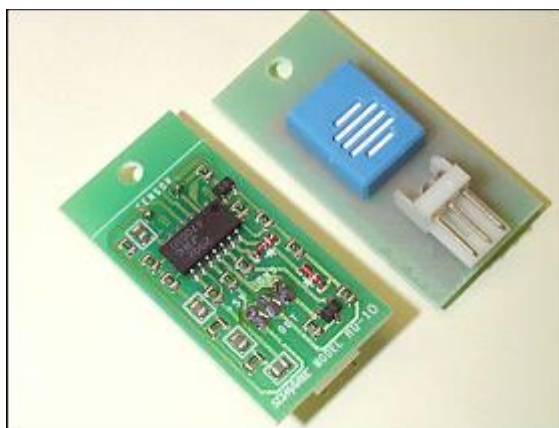


Rys. 5.6 Czujnik ciśnienia Telaire 6004 [29]

Do pomiaru wilgotności powietrza wykorzystać można sensor RHU1015. Jest układ wykorzystujący oporowy czujnik HS15P. Parametry układu przedstawiono w tabeli 5.7. Rysunek 5.7 przedstawia układ RHU1015.

Tab. 5.7 Parametry układu RHU1015 [26]

Parametr	Wartość
Napięcie zasilania	5±0,2 V DC
Warunki pracy – temperatura	0÷50°C
Warunki pracy – wilgotność	10÷100% RH
Maksymalny prąd	2 mA
Napięcie wyjściowe	1÷3 V DC
Dokładność 25÷90% RH @ 25°C	<±5% RH
inny zakres	<±10% RH
Typowy czas reakcji	5 min
Cena	~60 zł.



Rys. 5.7 Czujnik wilgotności RHU1015 [26]

Układ iButton DS1923 jest rejestratorem temperatury i wilgotności powietrza. Układ jest samowystarczalnym systemem pomiarowym, który mierzy temperaturę i/lub wilgotność i przechowuje wyniki pomiarów w chronionej części pamięci. Odczyty są wykonywane w zdefiniowanych przez użytkownika odstępach. Maksymalna ilość 8192 8-bitowych lub 4096 16-bitowych pomiarów wykonywanych jest w równych odstępach czasu od 1 sekundy do 273 godzin. Urządzenie może zostać zaprogramowane do wykonywania pomiarów zaraz po uruchomieniu, ze zdefiniowanym opóźnieniem lub po wywołaniu przez alarm (zmiana temperatury ponad/poniżej zdefiniowanego poziomu). Dostęp do pamięci układu DS1923 może być zabezpieczony hasłem. W tabeli 5.8 przedstawiono parametry rejestratora DS1923.

Tab. 5.8 Parametry układu iButton DS1923 [13]

Parametr	Wartość
Rozdzielczość pomiaru wilgotności	8-bitowa – 0,6% RH 12-bitowa – 0,04% RH
Rozdzielczość pomiaru temperatury	8-bitowa – 0,5°C

	12-bitowa – 0,0625°C
Dokładność pomiaru wilgotności	±5% RH
Dokładność pomiaru temperatury	±0,5°C w temperaturze -10°C÷65°C
Warunki pracy	-20°C÷85°C

6. Obsługa urządzeń 1-Wire

6.1. Wprowadzenie

Na rynku oprogramowania istnieją rozwiązania przeznaczone do obsługi urządzeń wykonanych w technologii 1-Wire. Producent układów, firma Maxim na swoich stronach internetowych udostępnia dla wszystkich tzw. SDK²⁹. Zawiera ono szeroką gamę przykładów implementacji obsługi urządzeń 1-Wire przez systemy komputerowe. Kody źródłowe, biblioteki systemowe dostępne są w różnych językach programowania (C, JAVA, .NET), wykorzystywać można je na różnych platformach sprzętowych i programowych. Firma Maxim opracowała „przeglądarkę” OneWireViewer, program napisany w języku JAVA, który umożliwia użytkownikowi sprawdzenie podstawowych parametrów urządzeń pracujących w jego sieci 1-Wire. Najbardziej popularnym zastosowaniem technologii 1-Wire jest dokonywanie pomiarów temperatury. Firma Maxim produkuje proste układy scalone, dzięki czemu na rynku powstało wiele urządzeń, wykorzystujących poszczególne układy. Dostępne są np. różnego rodzaju „stacje pogodowe”, które wykorzystują układy Maxim do zbierania danych, na temat aktualnych warunków atmosferycznych. Wykorzystywanie czujników 1-Wire w takich stacjach, w powiązaniu z układami od różnych producentów powoduje konieczność opracowania oprogramowania pod konkretny model takich narzędzi. Do współpracy z instalacjami własnej konstrukcji (np. sieć czujników w domu jednorodzinnym, monitor temperatur w chłodni itd.) należy posłużyć się materiałami od firmy Maxim i stworzyć autorski program. Do takich celów powstaje cała gama uniwersalnych programów, opracowana przez programistów-amatorów, przy czym programy te niejednokrotnie posiadają bogate możliwości. Jednym z takich programów jest fiński Lämpömittari (termometr), napisany przez Timo Sara-aho.

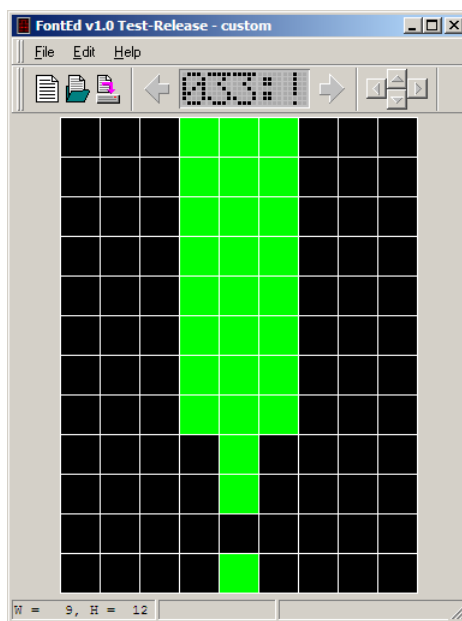
²⁹ SDK (ang. Software Development Kit) - zestaw narzędzi dla programistów niezbędny w tworzeniu aplikacji korzystających z danej biblioteki, pod dany system czy sprzęt.

6.2. Dostępne oprogramowanie realizujące komunikację sieciową MicroLAN

Oprogramowanie fińskiego programisty składa się z trzech modułów [18]:

- edytor znaków (ang. *FontEd*) prezentowanych na wirtualnym wyświetlaczu LCD³⁰,
- edytor wskaźników (ang. *Analog Configurator*), umożliwia tworzenie własnych wskaźników,
- moduł główny (ang. *Thermometer*) prezentacji i archiwizacji wyników pomiarów.

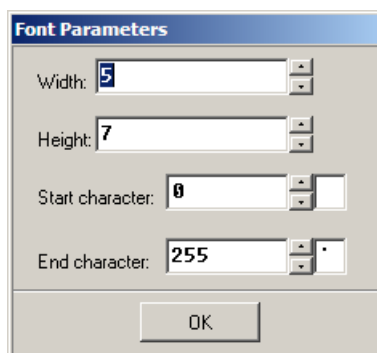
Należy zaznaczyć, że wszystkie pomiary w programie Lampomittari wykonane zostały przy użyciu układów (sterownika i czujników), wykorzystywanych w praktycznej części tej pracy. Za pomocą edytora znaków możemy utworzyć ich zestaw, dzięki któremu podczas działania głównego modułu, wyświetlane będą pomiary temperatury. Prezentacja wyników na pseudo-wyświetlaczu LCD jest jedną z kilku opcji, jakie posiada program Lampomittari. Interfejs programu przedstawia rysunek 6.1.



Rys. 6.1 Program FontEd w trybie edycji

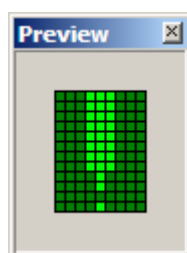
Po uruchomieniu programu należy wybrać tworzenie nowego zestawu lub edycja poprzednio zapisanego. W pierwszym przypadku należy wybrać rozmiar matrycy i zakres znaków (rys. 6.2).

³⁰ LCD (ang. Liquid Crystal Display) – wyświetlacz ciekłokrystaliczny.



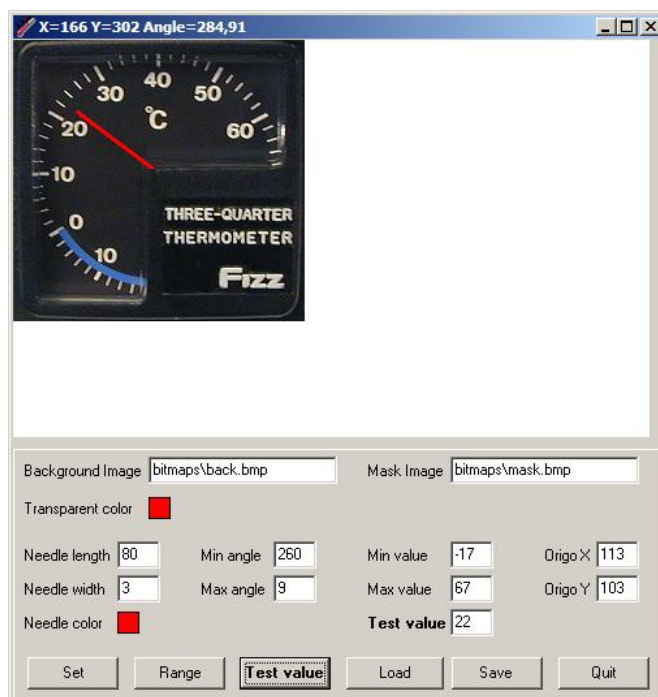
Rys. 6.2 Ustawienia matrycy edytowanego zestawu znaków w programie FontEd

Następnie za pomocą kursora myszki zaznaczamy, które punkty będą świecić na wyświetlaczu, co w rezultacie tworzy dany znak. Użytkownik ma do dyspozycji podstawową paletę edycji: kopiowanie, wklejanie, przesuwanie (za pomocą kursorów), czyszczenie, wypełnianie, odwracanie. W celu sprawdzenia, jak utworzony znak będzie wyglądał na wyświetlaczu, można posłużyć się podglądem (rys. 6.3). Obraz w oknie podglądu aktualizuje się na bieżąco, w czasie edycji kolejnych punktów.



Rys. 6.3 Podgląd tworzonego znaku w programie FontEd

Przy pomocy modułu Analog Configurator można stworzyć własny wskaźnik analogowy, który podobnie jak wyświetlacz LCD, prezentuje wyniki pomiarów temperatury. Wygląd takiego miernika to tarcza ze wskazówką. Do programu należy wczytać własnoręcznie utworzoną lub przykładową bitmapę, zawierającą skalę. Następnie należy ustawić wskazówkę, jej położenie reguluje się za pomocą współrzędnych X i Y. Ponadto skonfigurować należy kąt obrotu (0÷360 stopni) i zakres pomiaru (wartość minimalna i maksymalna). Program umożliwia również zmianę grubości, długości i koloru wskazówki. Do sprawdzenia działania utworzonego wskaźnika służy przycisk „Range”, który obróci wskazówkę w całym zadeklarowanym zakresie. Możliwe jest również sprawdzenie konkretnej wartości, przy użyciu pola „Test value” i przycisku z tą samą etykietą. Okno programu Analog Configurator przedstawia rysunek 6.4.



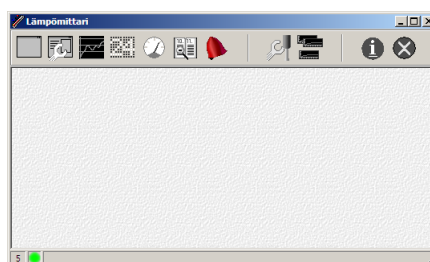
Rys. 6.4 Okno programu Analog Configurator

Po zakończeniu edycji miernika analogowego, można go zapisać. W celu odwołania się do utworzonego wskaźnika w module głównym należy, po wcześniejszym umieszczeniu, wybrać go z menu dostępnego pod prawym przyciskiem myszy (rys. 6.5).



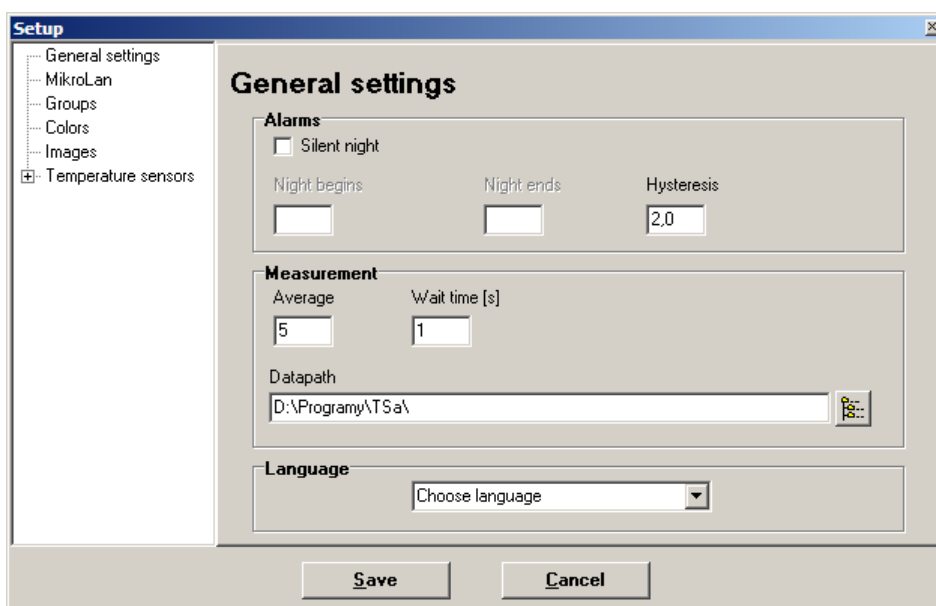
Rys. 6.5 Wybór wskaźnika w programie Lampomittari

Ostatnim i najważniejszym z elementów oprogramowania Lampomittari jest program Thermometer. Przed uruchomieniem programu należy podłączyć do komputera czujniki temperatury. Po uruchomieniu programu widać puste okno i pasek zawierający ikony (rys. 6.6).



Rys. 6.6 Okno główne programu Thermometer

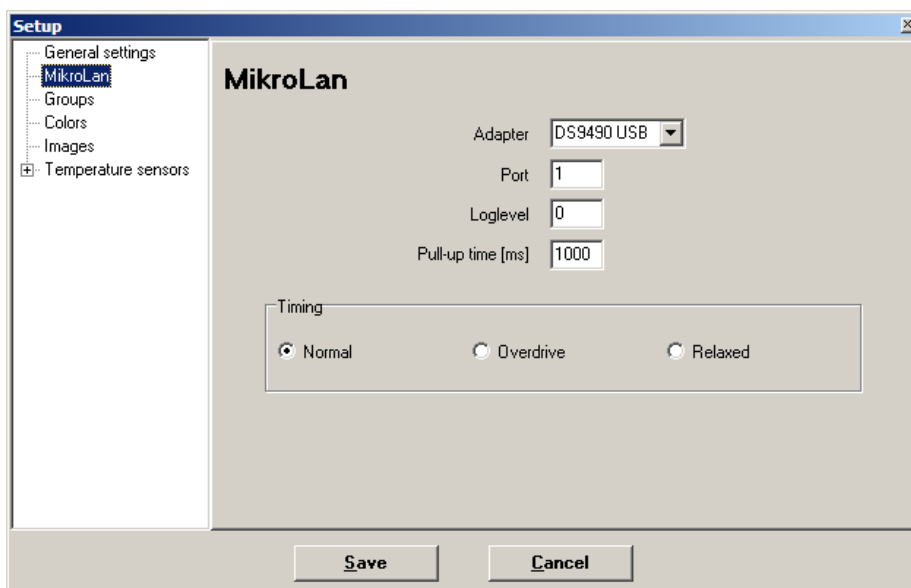
W celu zmiany ustawień, należy otworzyć okno konfiguracji (rys. 6.7).



Rys. 6.7 Ustawienia główne

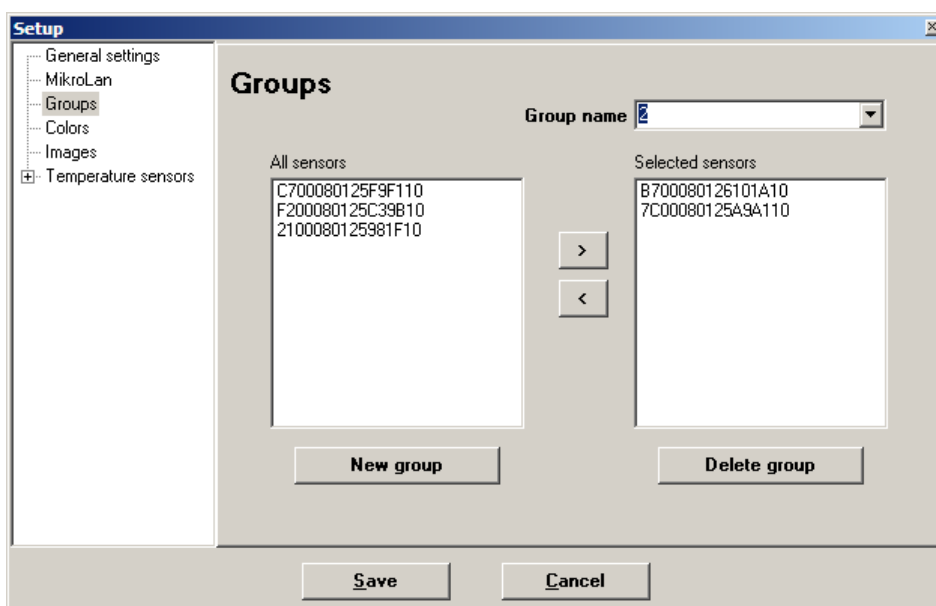
Na zakładce ustawień podstawowych, można dostosować: tryb nocny (kiedy alarmy będą wyciszone), ilość i odstęp pomiędzy kolejnymi pomiarami, ścieżkę dostępu do folderu w którym będą zapisywane wyniki pomiarów, język programu (w tym polski).

Zakładka MikroLan służy do konfiguracji sieci 1-Wire podłączonej do komputera użytkownika programu. W celu prawidłowego działania programu, należy wybrać odpowiedni adapter (interfejs) i numer portu do którego jest podłączony. Możliwe jest również ustawienie czasu ładowania czujnika, przed dokonaniem konwersji (ustawienie nie ma znaczenia, jeżeli czujniki używają zewnętrznego zasilania). Na rysunku 6.8 przedstawiono okno konfiguracji MikroLan.



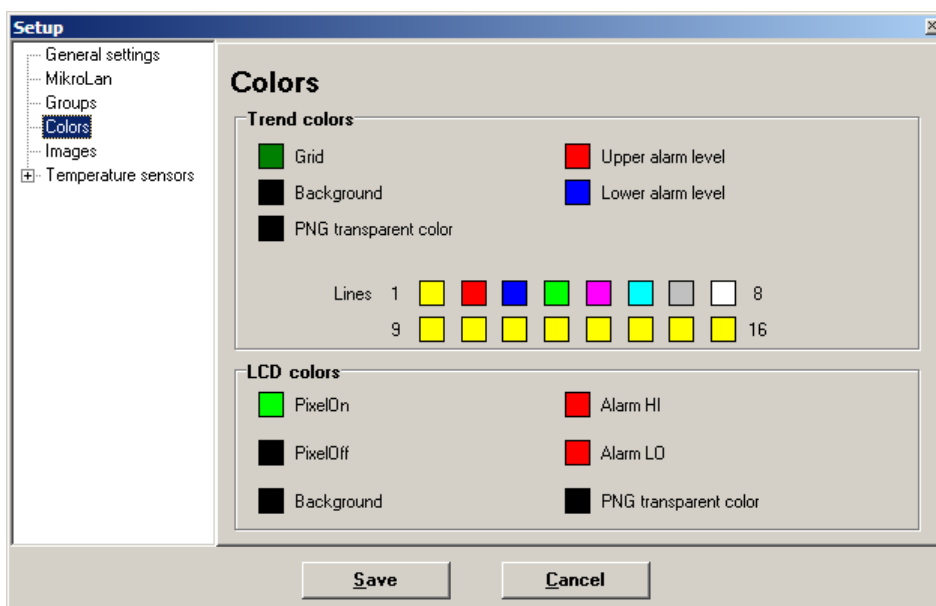
Rys. 6.8 Konfiguracja MikroLan w programie Thermometer

Przydatną funkcją programu jest grupowanie czujników. Jeżeli w danym punkcie sieci występuje kilka sensorów, warto jest gromadzić ich pomiary w ramach jednego zespołu. Jeden czujnik może przynależeć do kilku grup. Okno ustawień grup przedstawia rysunek 6.9.



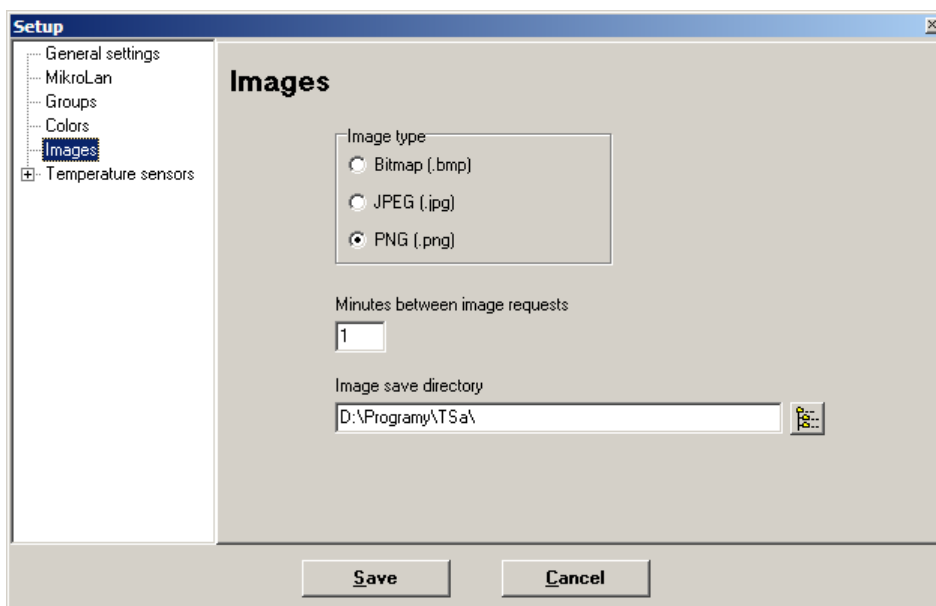
Rys. 6.9 Konfiguracja grup czujników w programie Thermometer

Program Thermometer umożliwia użytkownikowi dopasowanie kolorów wyświetlanych pomiarów do jego potrzeb. Przydatne może okazać się zdefiniowanie kolorów, kiedy czujniki wykazują przekroczenie zakresów alarmowych (minimalnego i maksymalnego). Barwy wybrać można dla wykresów (historii pomiarów i trendu) i wyświetlacza LCD. Konfigurację kolorów przedstawia rysunek 6.10.



Rys. 6.10 Ustawienia kolorów w programie Thermometer

Gromadzenie danych pomiarowych umożliwia ich późniejszą analizę. Program Lampomittari obok gromadzenia danych liczbowych, daje użytkownikowi możliwość archiwizacji stanów wskaźników. Wykorzystywane są obrazy typu: mapa bitowa, skompresowany obraz JPG³¹ lub PNG³² (wraz z kanałem przezroczystości). Użytkownik programu może wybrać odstęp pomiędzy kolejnymi zapisami obrazów i miejsce ich przechowywania. Należy zaznaczyć, że kolejno zapisywane obrazy, nadpisują poprzednie. Okno ustawień przedstawia rysunek 6.11.



Rys. 6.11 Ustawienia obsługi obrazów w programie Thermometer

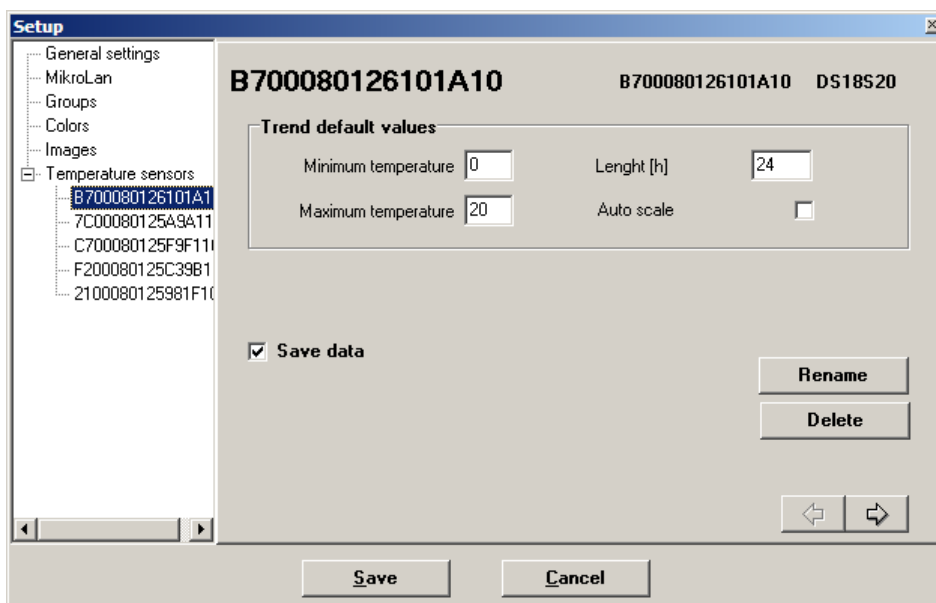
Najbardziej rozbudowaną część programu Thermometer stanowi konfiguracja sensorów temperatury. Każdy czujnik konfiguruje się na jednej z czterech zakładek: ustawienia wartości alarmów, konfiguracja alarmów, wykrywanie trendu, korekcja błędów. Na pierwszej planszy zdefiniować można niezależnie alarm minimum i maksimum, zmienić fabryczny unikatowy identyfikator sensora (rys 6.12). Druga zakładka pozwala określić dźwięk alarmów oraz przypisanie do niego operacji. Możliwe jest uruchomienie wybranej przez użytkownika aplikacji lub wykonanie pliku wsadowego (np. wysłanie wiadomości SMS³³). Konfigurację alarmów przedstawia rysunek 6.13. Na podstawie zebranych wcześniej wyników pomiarów możliwe jest wykrywanie trendu zmiany temperatury. Dzięki temu użytkownik może zdefiniować operację, która zostanie wykonana w przypadku wystąpienia ciągłego jej wzrostu lub spadku (np. wyłączenie ogrzewania w przypadku wzrostu temperatury zewnętrznej). Okno obsługi wykrywania trendów przedstawia rysunek 6.14.

³¹ JPG, JPEG (ang. Joint Photographic Experts Group) – stratny format kompresji obrazów.

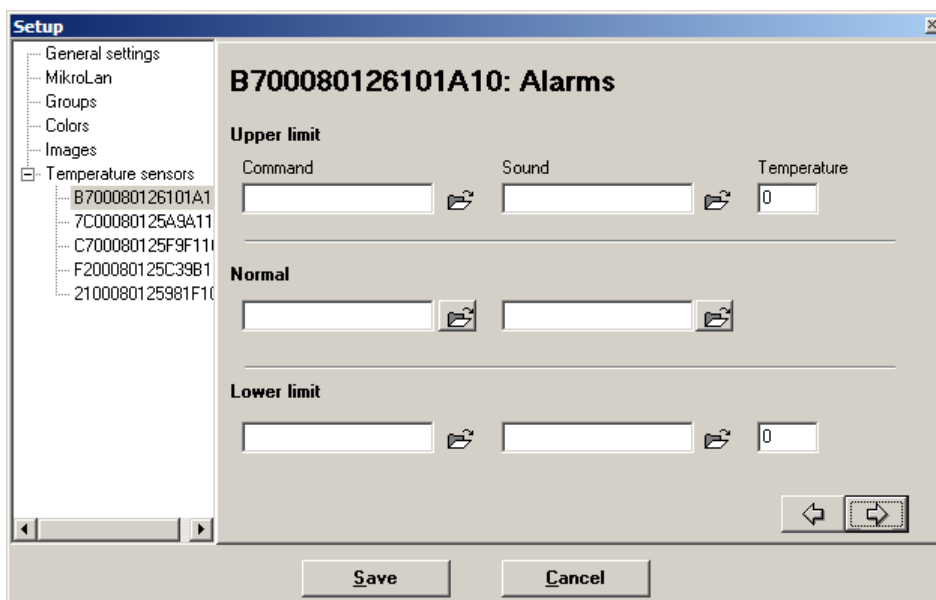
³² PNG (Portable Network Graphics) – bezstratny format kompresji obrazów.

³³ SMS (ang. Short Message Service) – usługa przesyłania wiadomości tekstowych za pomocą telefonów komórkowych.

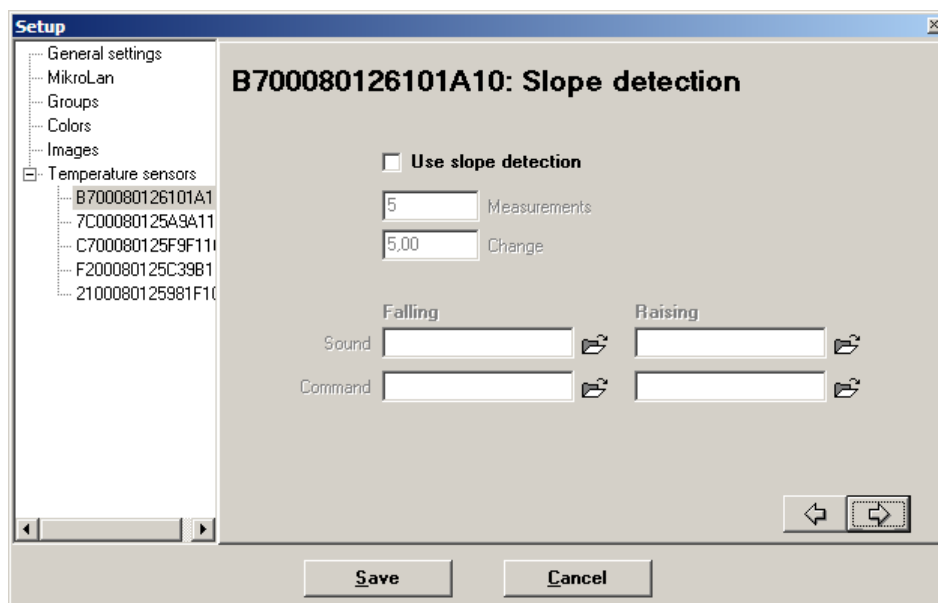
Czujniki temperatury 1-Wire dają możliwość pracy nawet z 12 bitową rozdzielczością. Funkcja ta jest przydatna, kiedy istotna jest obserwacja wahań setnych części temperatury. Charakterystyka pracy sensorów umożliwia łatwą korekcję błędów w zakresie dokonywanych pomiarów. Błąd pomiaru czujników DS opisany jest zależnością kwadratową, której parametry daje się w łatwy sposób wyznaczyć. Parametry określają położenie minimum funkcji błędu i jej kształt. Szczegóły opisano w podrozdziale 4.2.5 tej pracy. Parametry korekcji błędów przedstawia rysunek 6.15.



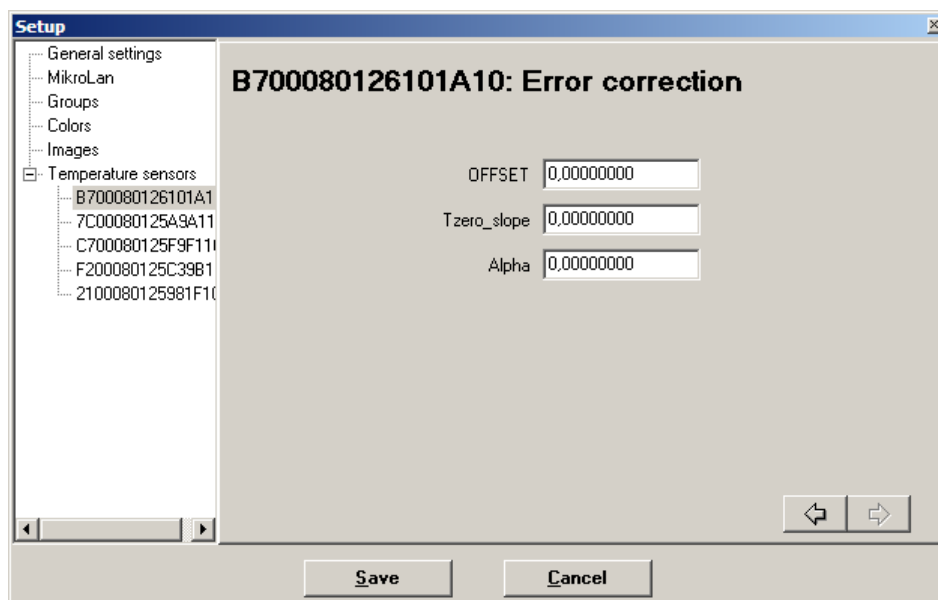
Rys. 6.12 Konfiguracja sensorów w programie Thermometer



Rys. 6.13 Ustawienia alarmów czujników w programie Thermometer



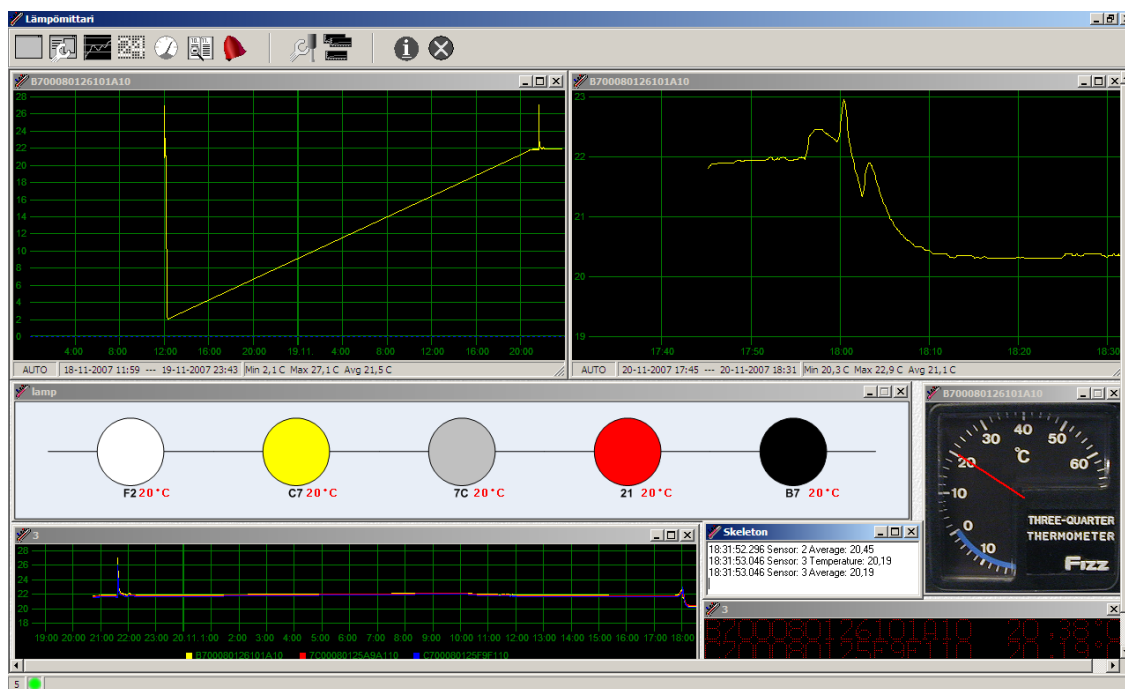
Rys. 6.14 Wykrywanie trendu zmiany temperatury w programie Thermometer



Rys. 6.15 Korekcja błędów pomiarów czujników temperatury w programie Thermometer

W trakcie pracy z programem Thermometer, na ekranie prezentowane są wyniki pomiarów temperatury w zdefiniowany przez użytkownika sposób. Przykładowy sposób prezentacji wybranych pomiarów podczas pracy programu pokazuje rysunek 6.16. Zaczynając od góry i od lewej strony widać kolejno: wykres historii, wykres trendu, wykres dowolny, miernik analogowy, podgląd logu, wyświetlacz LCD. Każdy z elementów konfiguruje się odrębnie za pomocą menu dostępnego pod prawym przyciskiem myszy. Każde okno można dowolnie skalować (z wyłączeniem wyświetlacza LCD i wykresu dowolnego) i rozmieszczać na ekranie. Sposób rozmieszczenia wszystkich wskaźników jest zapamiętywany przed

zamknięciem aplikacji. Okno programu można „schować” do obszaru systemowego (obok zegarka), za pomocą kombinacji CTRL i przycisku minimalizowania.



Rys. 6.16 Okno programu Thermometer z przykładowymi wskaźnikami

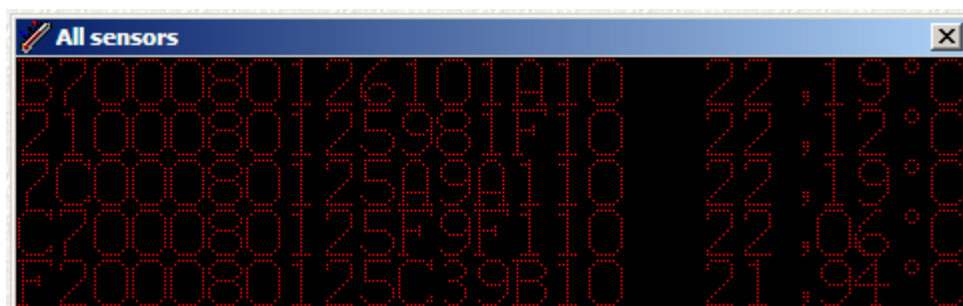
Pierwszym z grupy wskaźników jest wskaźnik analogowy. Tworzony jest w odrębnym programie Analog Configurator, prezentuje wyniki pomiaru jednego sensora. Dodatkową opcją jest możliwość okresowego zapisywania stanu miernika. Wygląd miernika przedstawia rysunek 6.17.



Rys. 6.17 Przykładowy wskaźnik analogowy programu Thermometer

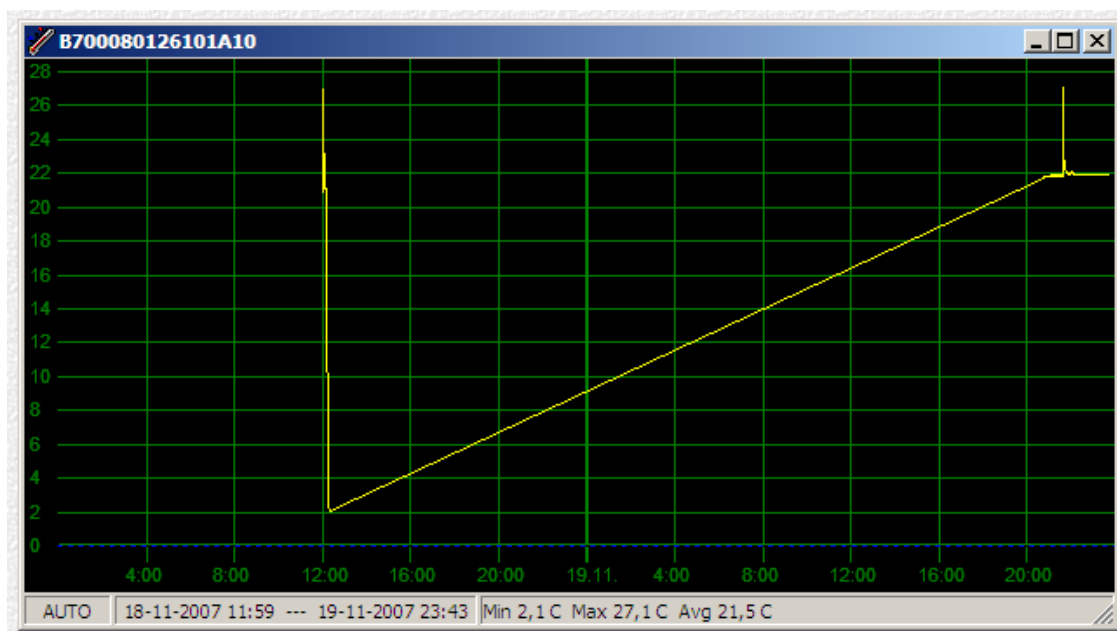
Drugim ze wskaźników jest wyświetlacz LCD. W oknie programu Lampomittari na utworzonym wyświetlaczu można umieścić zdefiniowaną wcześniej grupę czujników temperatury. Użytkownik programu może wybrać, czy temperatura prezentowana ma być w stopniach Celsjusza lub Farenheita. Do wyboru są dwa rozmiary okna, duży lub mały,

różniące się rozdzielczością. Wielkość znaków zdefiniowana jest poprzez użytą czcionkę, którą w miarę potrzeby można stworzyć w programie FontEd. Stan wyświetlacza można eksportować do pliku, podobnie jak w przypadku wskaźnika analogowego. Przykład wyświetlacza LCD umieszczony jest na rysunku 6.18.



Rys. 6.18 Wyświetlacz LCD prezentujący pomiar z pięciu czujników temperatury

Obok prezentacji wyników aktualnej temperatury, istotne jest również zbieranie i prezentacja danych zebranych na przestrzeni pewnego okresu czasu. Do tego celu w programie Thermometer można wykorzystać wykres historii. W celu wyświetlenia przebiegu temperatury w zadanym okresie dla konkretnego czujnika, należy wybrać ikonę książki z menu programu. W tym miejscu należy wskazać interesujący sensor i interesujący zakres czasu. Po ustawieniu wszystkich parametrów, na ekranie zobaczyć można wykres, przykład którego pokazuje rysunek 6.19.

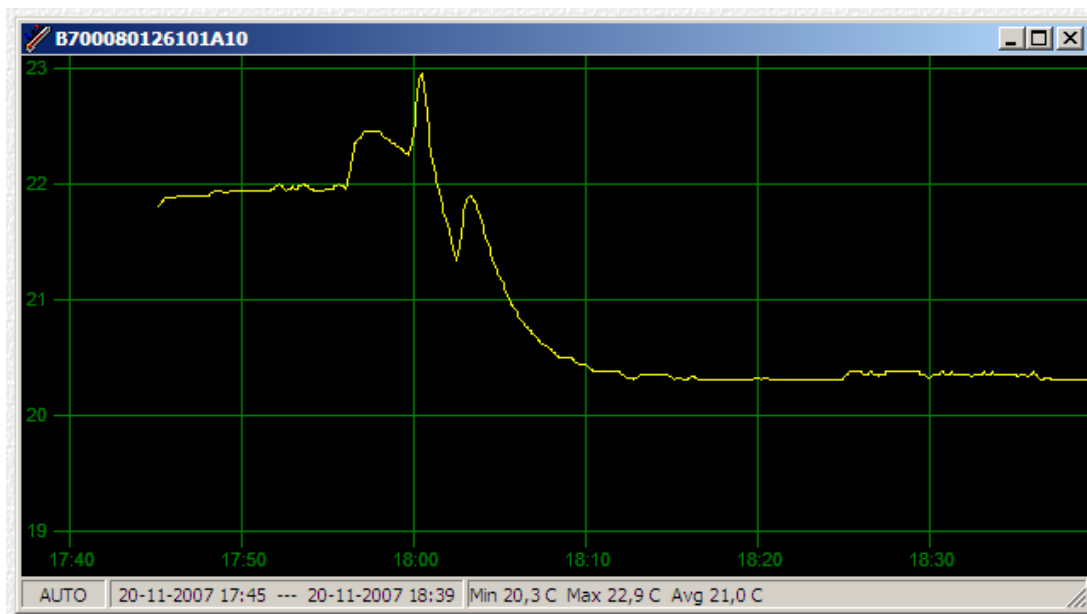


Rys. 6.19 Wykres historii czujnika temperatury

Wstawiony wykres można dodatkowo przeskalować według zadanych kryteriów. Do dyspozycji użytkownika jest skalowanie czasu i temperatury. Jednostka czasu zawiera się w zakresie od dziesięciu minut do dwóch tygodni, a okresy trendu od jednej godziny do

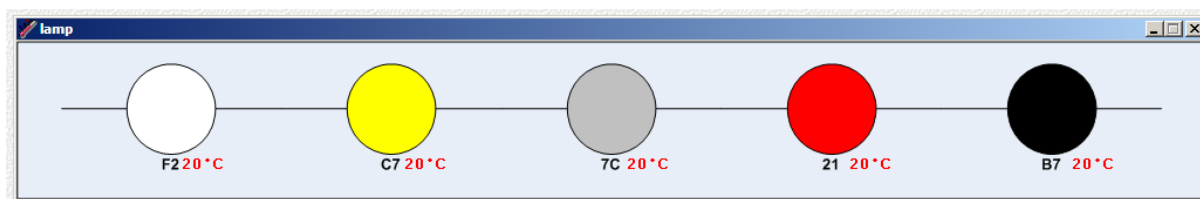
dwóch miesięcy. Skalowanie temperatury dostępne jest z pośród wartości: 1, 2, 5, 10 stopni. W razie potrzeby możliwe jest wydrukowanie wykresu.

Innym rodzajem wykresu jest wykres trendu. Od poprzedniego różni go możliwość prezentacji wyników dla jednego bądź kilku czujników (przydzielonych do grup). Wykres trendu przedstawia rysunek 6.20.



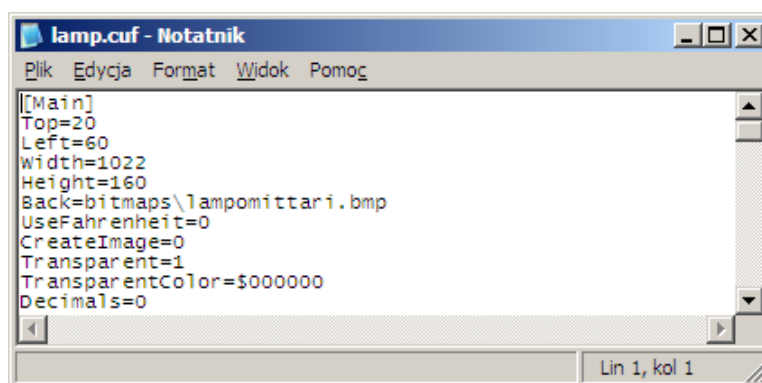
Rys. 6.20 Wykres trendu wybranego sensora

Ciekawą funkcją programu jest tworzenie własnych wskaźników, którego przykład przedstawia rysunek 6.21. Największym minusem tego narzędzia jest konieczność jego stworzenia w pliku tekstowym, przy użyciu określonych parametrów. Składnia pliku konfiguracyjnego opisana jest w pliku tekstowym „plugins.txt” położonym w katalogu głównym programu Lampomittari.



Rys. 6.21 Wskaźnik prezentujący pomiar z pięciu czujników temperatury

Poniżej znajduje się zawartość pliku wskaźnika, przedstawionego na rysunku 6.21.



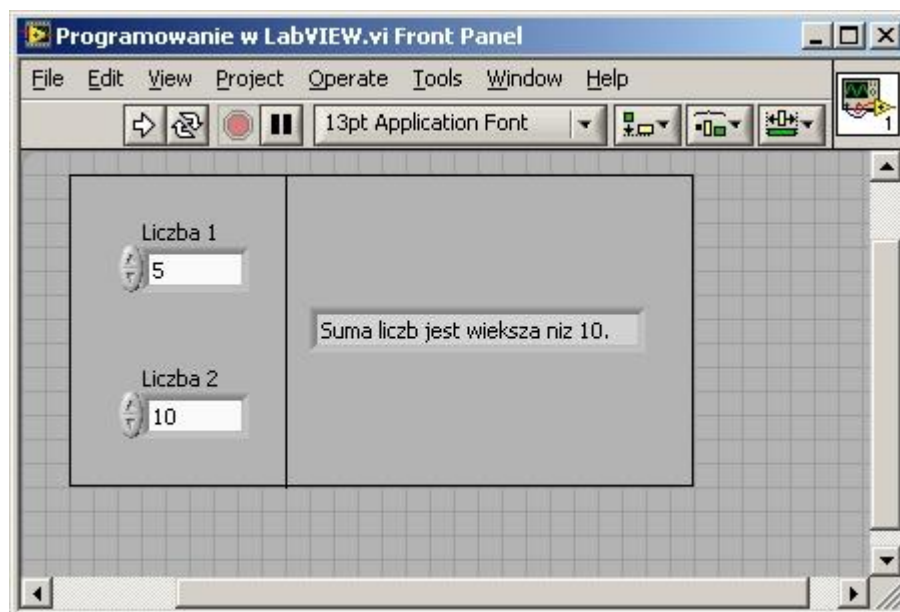
Rys. 6.22 Listing pliku utworzonego wskaźnika

Podsumowując program Lampomittari jest ciekawą propozycją na rynku programów obsługujących czujniki 1-Wire. Pomimo swoich niedociągnięć, daje użytkownikowi potężne narzędzie, służące do akwizycji i archiwizacji pomiarów dokonanych przy użyciu tych czujników. Program oferuje atrakcyjną formę prezentacji, nie zabiera więcej zasobów komputera, niż 5 MB pamięci operacyjnej, w zależności od ilości umieszczonych wskaźników. Największą zaletą programu jest możliwość wykorzystania sensorów do granic ich możliwości. Dzięki korekcji błędów pomiarowych „zestaw” Lampomittari to prosty i skuteczny monitor temperatury.

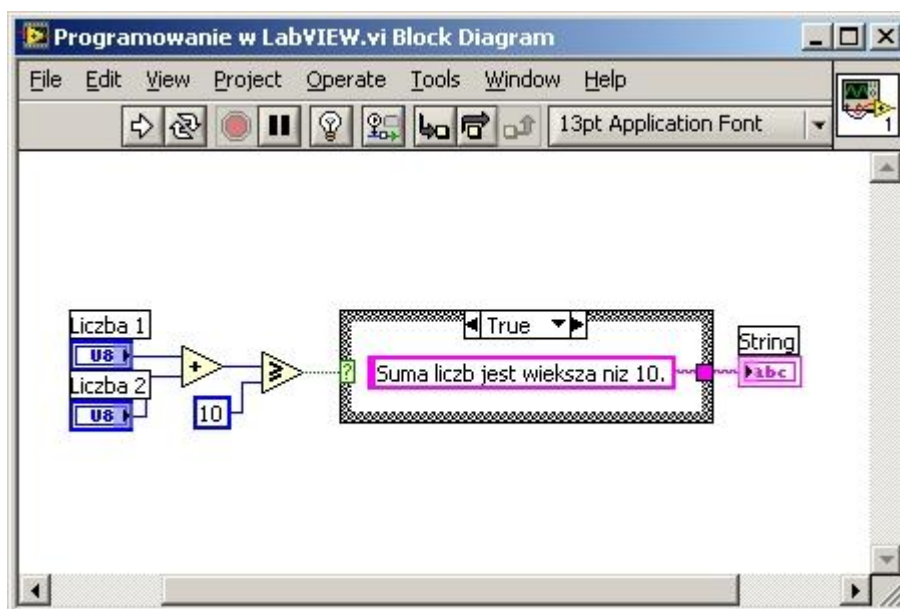
6.3. Opis środowiska LabVIEW

LabVIEW jest graficznym środowiskiem programowania. W klasycznych środowiskach tekstowych, program składa się z kolejnych linii instrukcji, nazywanych kodem programu. Programowanie graficzne opiera się o umieszczanie zmiennych, procedur i struktur w postaci ikon na diagramie i odpowiednie ich łączenie. Programy tworzone w środowisku LabVIEW imitować mogą rzeczywiste urządzenia, stąd nazywane są instrumentami wirtualnymi (ang. *virtual instruments*, w skrócie *vi*), a język programowania nosi nazwę „G”.

Każdy program stworzony w środowisku LabVIEW składa się z pary okien: panelu frontowego (ang. *front panel*) i diagramu blokowego (ang. *block diagram*). Panel służy do komunikacji z użytkownikiem. Za jego pomocą wprowadzane są dane i prezentowane wyniki działania aplikacji. Diagram odpowiada za wykonywanie wszystkich funkcji i przekazywanie wyników na panel frontowy. Ważnym elementem programu jest ikona, która reprezentuje go na diagramie, dzięki czemu możliwe jest użycie programu w postaci procedury (wewnątrz innego *vi*’ja). Przykładowy panel frontowy przedstawiony jest na rysunku 6.23, a diagram blokowy na rysunku 6.24.



Rys. 6.23 Przykładowy wygląd panelu frontowego

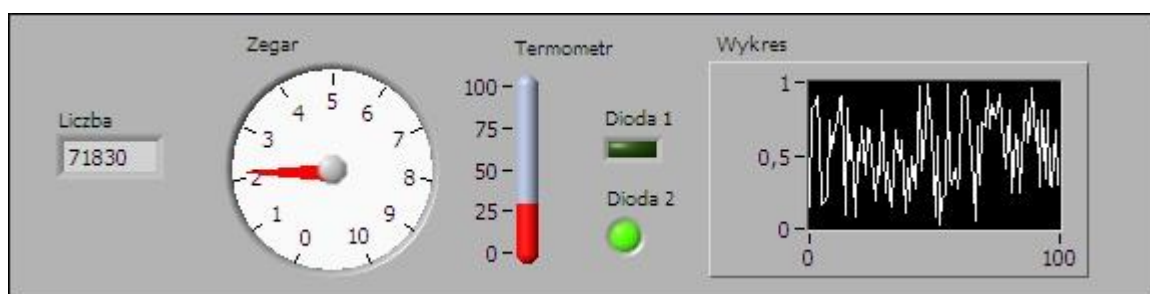


Rys. 6.24 Przykładowy wygląd diagramu blokowego

Na panelu frontowym umieszcza się kontrolki (ang. *controls*) i wskaźniki (ang. *indicators*), które reprezentują odpowiednio wejścia i wyjścia programu. Przykładem kontrolki są: pokrętki, przyciski, suwaki. Przykładem wskaźników są: wykresy, diody, zegary. Na rysunkach 6.25 i 6.26 przedstawiono kolejno przykłady kontrolki i wskaźników.



Rys. 6.25 Przykład kontrolki umieszczanych na panelu frontowym programu

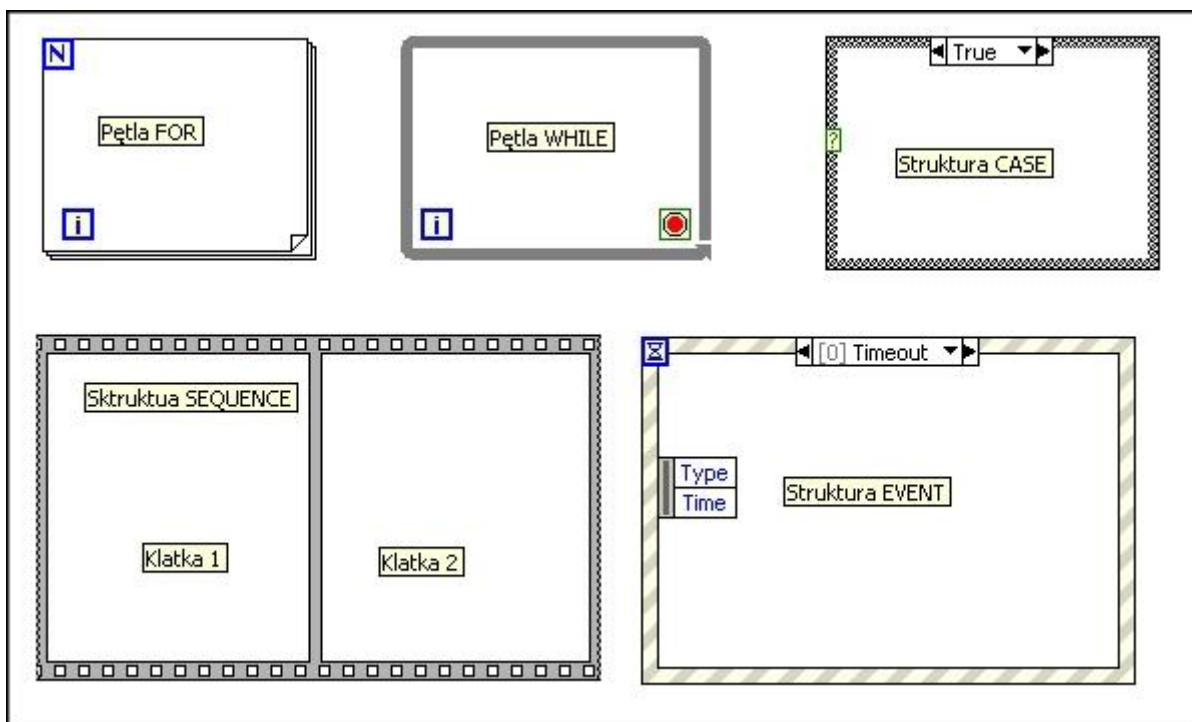


Rys. 6.26 Przykład wskaźników umieszczanych na panelu frontowym programu

Diagram blokowy programu zawiera terminale, funkcje, struktury i łączące je ze sobą przewody. Terminale pośredniczą w wymianie danych między diagramem a panelem. Posiadają one określony typ i właściwość (zapis do – odpowiada wskaźnikowi, odczyt z – odpowiada kontrolce). Funkcje są to programy, które pobierają wartości, wykonują operacje i zwracają wyniki. W środowisku LabVIEW wyróżniamy dwa rodzaje programów, pierwotne i tzw. subVI. Pierwotnych funkcji, w przeciwieństwie do subVI, nie można edytować, nie posiadają panelu frontowego ani diagramu i wykonują najczęściej podstawowe operacje (numeryczne, logiczne itp.). SubVI to program zbudowany z programów pierwotnych lub/i z innych podprogramów (subVI określa się również jako podprogramy). Podprogramy można przeglądać (panel i diagram) i w razie potrzeby edytować. Struktury umożliwiają wykonywanie operacji w pętlach iteracyjnych, bądź warunkowych. Struktury posiadają pewne domyślne terminale sterujące (np. licznik pętli) i takie, które służą do przekazywania danych z kontrolki (terminale wejściowe) i do wskaźników (lub innych struktur czy funkcji). Na diagramie blokowym można stosować poniższe struktury:

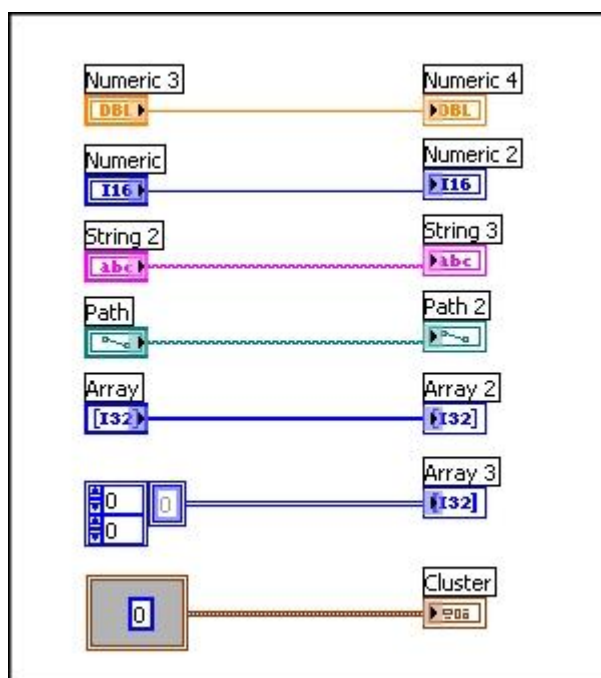
- pętla FOR – wykonuje się zadaną ilość razy,
- pętla WHILE – wykonuje się ciągle aż do spełnienia warunku stop,
- struktura CASE – wykonuje operacje w przypadku spełnienia określonego warunku,
- struktura SEQUENCE – sekwencyjne wykonywanie operacji w określonej kolejności,
- struktura EVENT – wykonuje operacje w reakcji na określone zdarzenie użytkownika.

Struktury stosowane w środowisku LabVIEW przedstawia rysunek 6.27.



Rys. 6.27 Struktury i pętle środowiska LabVIEW

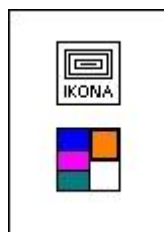
Przepływ danych na diagramie reprezentowany jest przez przewody. Łączą one funkcje z kontrolkami i wskaźnikami. W zależności od typu danych przewody posiadają różne kolory, grubość i styl (rys. 6.28). Połączenia mają zawsze jedno źródło, ale mogą być rozdzielone i doprowadzone do wielu celów (np. do wskaźnika lub kolejnej funkcji).



Rys. 6.28 Przykładowe przewody łączące obiekty na diagramie blokowym

W celu zapisania aplikacji w postaci podprogramu (subVI) należy zdefiniować połączenia. W celu zdefiniowania połączeń, należy wybrać opcję edycji połączeń (ang. *show*

connectors) w obszarze ikony panelu frontowego. Następnie za pomocą myszki należy połączyć kontrolki i wskaźniki z terminalami. Odpowiednio kontrolki będą parametrami wejściowymi podprogramu, a wskaźniki wartościami zwracanymi przez program. Przykład ikony wraz ze zdefiniowanymi połączeniami przedstawia rysunek 6.29.



Rys. 6.29 Ikona programu subVI ze zdefiniowanymi połączeniami

Funkcje dostępne w środowisku LabVIEW pogrupowane są na palecie, dostępnej z poziomu diagramu blokowego. W trakcie budowy aplikacji, wybrane z palety funkcje umieszcza się na diagramie. Wygląd palety może różnić się, w zależności od używanej wersji środowiska LabVIEW. Na rysunku 6.30 przedstawiono przykładowy wygląd palety funkcji.



Rys. 6.30 Paleta funkcji środowiska LabVIEW

Na poszczególnych paletach znajdują się funkcje:

- *Measure I/O* – obsługa kart pomiarowych NI,

- *Instrument I/O* – obsługa urządzeń architektury VISA³⁴,
- *Mathematics* – analiza matematyczna, algebra, statystyka itp,
- *Sig Processing* – generowanie, analiza i filtracja sygnałów,
- *Data Comm* – komunikacja między aplikacjami,
- *Connectivity* – narzędzia .NET, ActiveX, porty we/wy,
- *Express* – popularne procedury pomiarowe wymagające minimum konfiguracji.

Paleta *Programming* zawiera następujące funkcje:

- *Structures* – struktury i funkcje,
- *Array* – operacje na tablicach,
- *Cluster* – operacje na zbiorach (rekordach),
- *Numeric* – operacje arytmetyczne, trygonometryczne, logarytmiczne itp;
- *Boolean* – operacje logiczne,
- *String* – operacje na ciągach znaków i ścieżkach,
- *Comparison* – operacje porównawcze,
- *Timing* – operacje związane z obsługą czasu,
- *Dialog & User Interface* – komunikacja z użytkownikiem, obsługa błędów,
- *File I/O* – operacje na plikach i katalogach.

Paletę *Programming* przedstawiono na rysunku 6.31.



Rys. 6.31 Paleta podstawowych funkcji do budowy aplikacji w LabVIEW

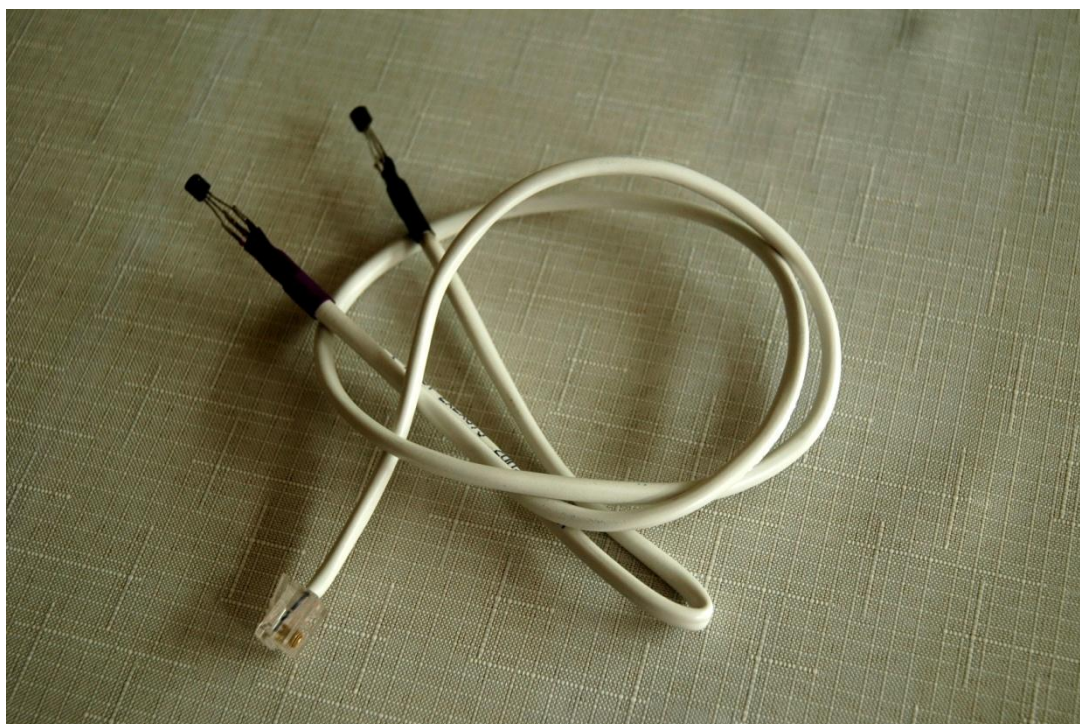
³⁴ VISA – ang. Virtual Instrument Software Architecture

6.4. Obsługa układów 1-Wire w środowisku LabVIEW

Opisywany w pracy układ pomiarowy składa się z następujących elementów:

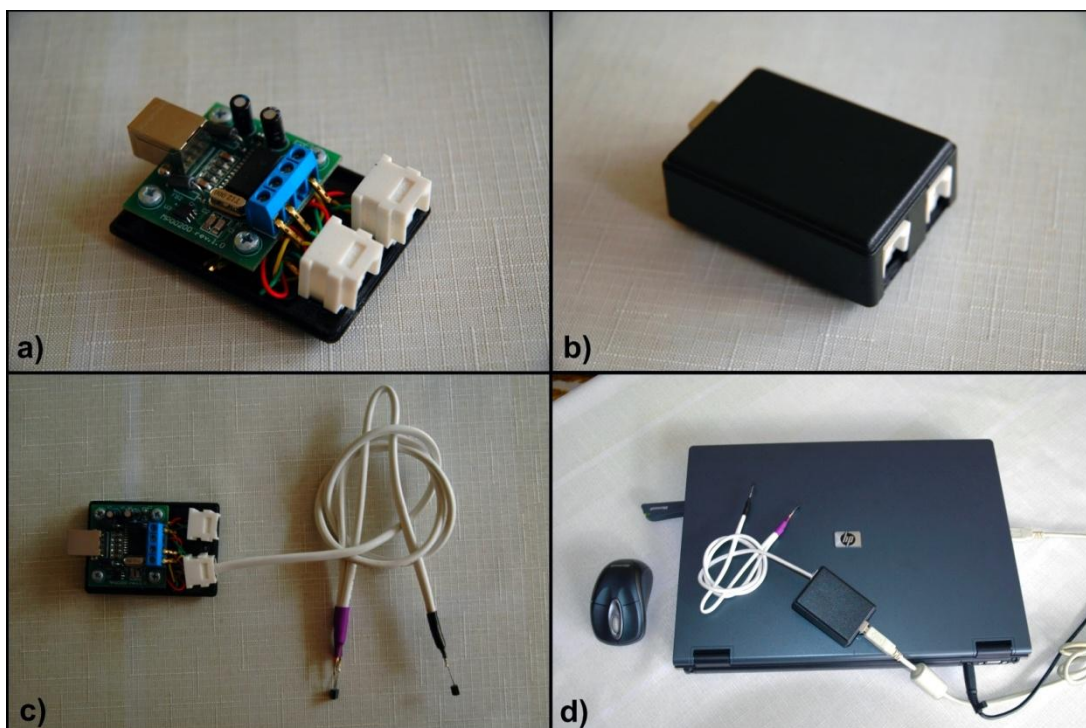
- czujników temperatury DS18B20,
- konwertera USB÷1-Wire MP00200,
- komputera wyposażonego w środowisko LabVIEW.

Na rysunku 6.32 przedstawiono parę czujników temperatury. Układy przylutowane są równoległe do przewodu telefonicznego, zakończonego wtykiem RJ-11. Parametry układów DS18B20 omówiono w rozdziale 5.2.2.



Rys. 6.32 Układ czujników DS18B20 wykorzystywanych w pracy

Do połączenia czujników temperatury z komputerem, wykorzystywany jest układ firmy Meraprojekt – MP00200. Jest to płytko drukowana, wyposażona w układ DS2490, gniazdo USB (do połączenia z komputerem) i listwy montażowe ARK (do połączenia z układami 1-Wire). Moduł jest w pełni zgodny z konwerterem DS9490 firmy Dallas Maxim. W celu zabezpieczenia, płytkę umieszczono w pudełku (dostępne w sklepach elektronicznych) i wyposażono w 2 gniazda telefoniczne (wymontowane z gniazdka telefonicznego). W ofercie firmy Meraprojekt dostępny jest również moduł MP00201, który jest zmodyfikowaną wersją modułu MP00200, wyposażony w obudowę, przewód USB i 2 gniazda telefoniczne. Na rysunku 6.33 przedstawiono wykorzystywany w pracy konwerter MP00200.



Rys. 6.33 Moduł MP00200 a) bez pokrywy, b) w zamkniętej obudowie, c) z podłączonymi czujnikami temperatury, d) w połączeniu z komputerem

Producent układów 1-Wire z pomocą pakietów SDK, daje programistom możliwość pisania aplikacji w dogodnym dla nich języku (patrz rozdział 6.1). W niniejszej pracy, do komunikacji wykorzystano API³⁵ *TMEX*. W LabVIEW wykorzystywana jest biblioteka *ibfs32.dll*, która dostarczona jest przez producenta układów 1-Wire w trakcie instalacji sterowników. Poszczególne funkcje wywoływane są w środowisku LabVIEW bezpośrednio z biblioteki, za pomocą bloku *Call Library Function Node*. Przy pisaniu programu wykorzystano elementy napisanej już aplikacji, dostępnej na stronach *National Instruments* (producenta LabVIEW) pod adresem <http://decibel.ni.com/content/docs/DOC-1092>.

6.5. Opis aplikacji

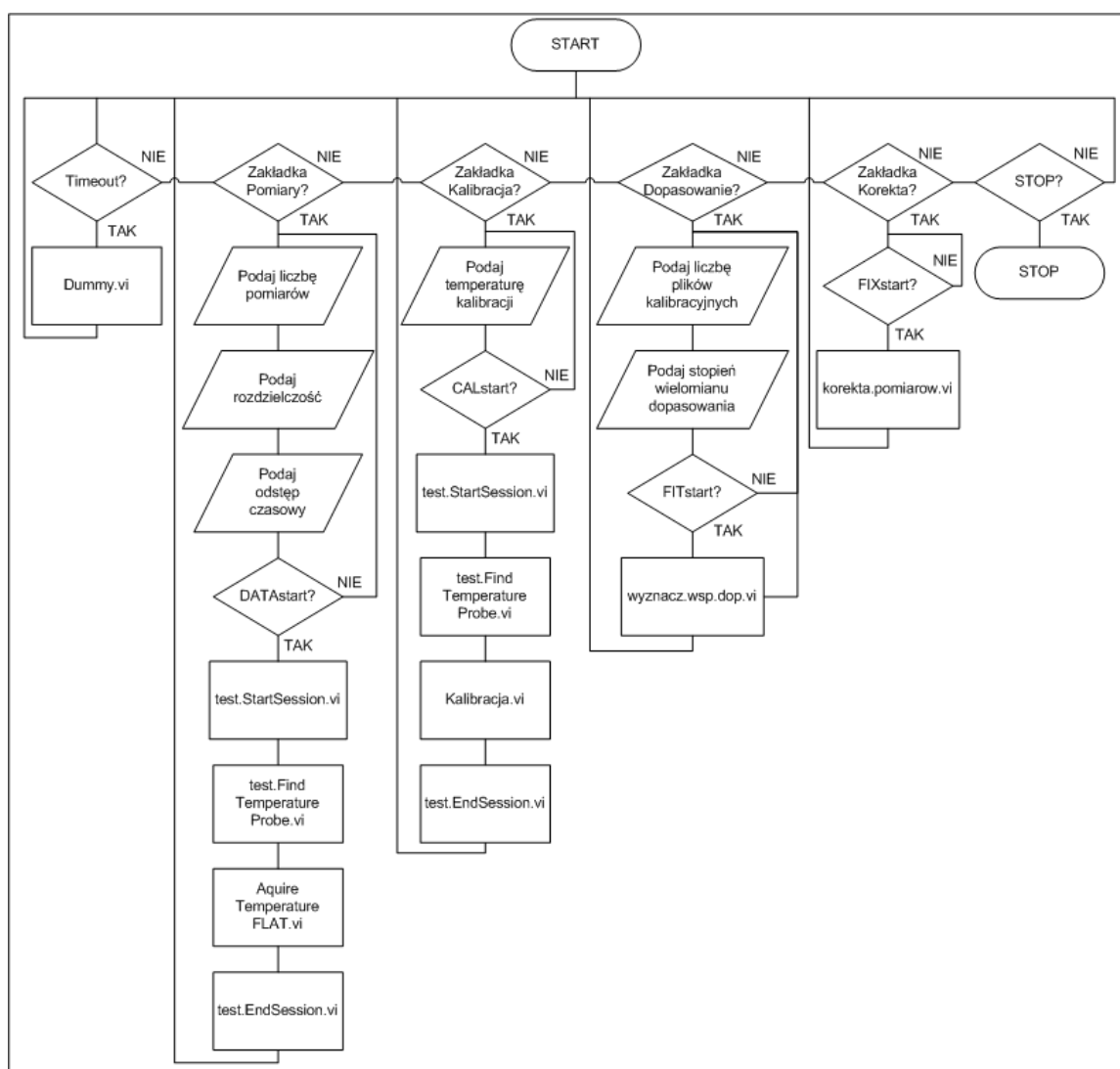
Oprogramowanie stworzone w środowisku LabVIEW do obsługi czujników temperatury 1-Wire, składa się z czterech modułów:

- pomiary,
- kalibracja,
- dopasowanie,
- korekta.

³⁵ API (ang. Application Programming Interface) – specyfikacja funkcji, procedur i interfejsów wykorzystywana do tworzenia aplikacji.

Moduł pomiarowy służy do wykonywania „surowych” pomiarów. Użytkownik programu określa liczbę pomiarów, odstęp między pomiarami (podawany w sekundach) i rozdzielczość. Wybranej rozdzielczości 9, 10, 11, 12 bitów odpowiadają przyrosty 0,5 °C, 0,25 °C, 0,125 °C, 0,0625 °C. Moduł kalibracyjny służy do wykonywania pomiarów, które za pomocą modułu dopasowania umożliwiają wyznaczenie współczynników dopasowania (patrz rozdział 4.2.5). W celu wykonania kalibracji, wszystkie czujniki jednocześnie umieszcza się w pobliżu źródła ciepła o stałej, znanej temperaturze. Po uruchomieniu przez użytkownika programu wykonywanych jest sto serii pomiarowych a wyniki zapisywane są do pliku. Moduł dopasowania służy do wyznaczenia współczynników wielomianu dopasowania. W zależności od ilości wczytanych plików kalibracyjnych, możliwe jest wyznaczenie współczynników wielomianu 0, 1, lub 2 stopnia. Moduł korekty pozwala poprawić „surowe” pomiary za pomocą współczynników wyznaczonych z pomocą modułu dopasowania.

Schemat blokowy aplikacji *PKDK.vi* przedstawia rysunek 6.34.



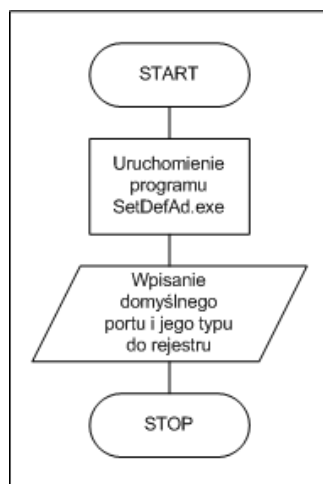
Rys. 6.34 Schemat blokowy programu *PKDK.vi*

Program główny w zależności o wybranego przez użytkownika modułu (dostępne na odpowiednich zakładkach) uruchamia odpowiednie procedury. Ponadto, w strukturze *EVENT* zdefiniowana jest ramka *Timeout*, która wykonuje się cyklicznie, gdy użytkownik nie wykonał żadnego ze zdefiniowanych zdarzeń (wciśnięcie przycisku). W ramce tej wykonuje się jeden pomiar a jego wynik rysowany jest na wykresie w osobnym oknie (program *dummy.vi*). Umożliwia to użytkownikowi podgląd aktualnej temperatury i uruchomienie programu, kiedy temperatura w czujnikach się ustabilizuje (bezwładność czujników DS18B20). Protokół 1-Wire wymaga wykonania odpowiednich operacji w określonej kolejności. Niektóre procedury wywoływane są wielokrotnie i omówione będą tylko raz.

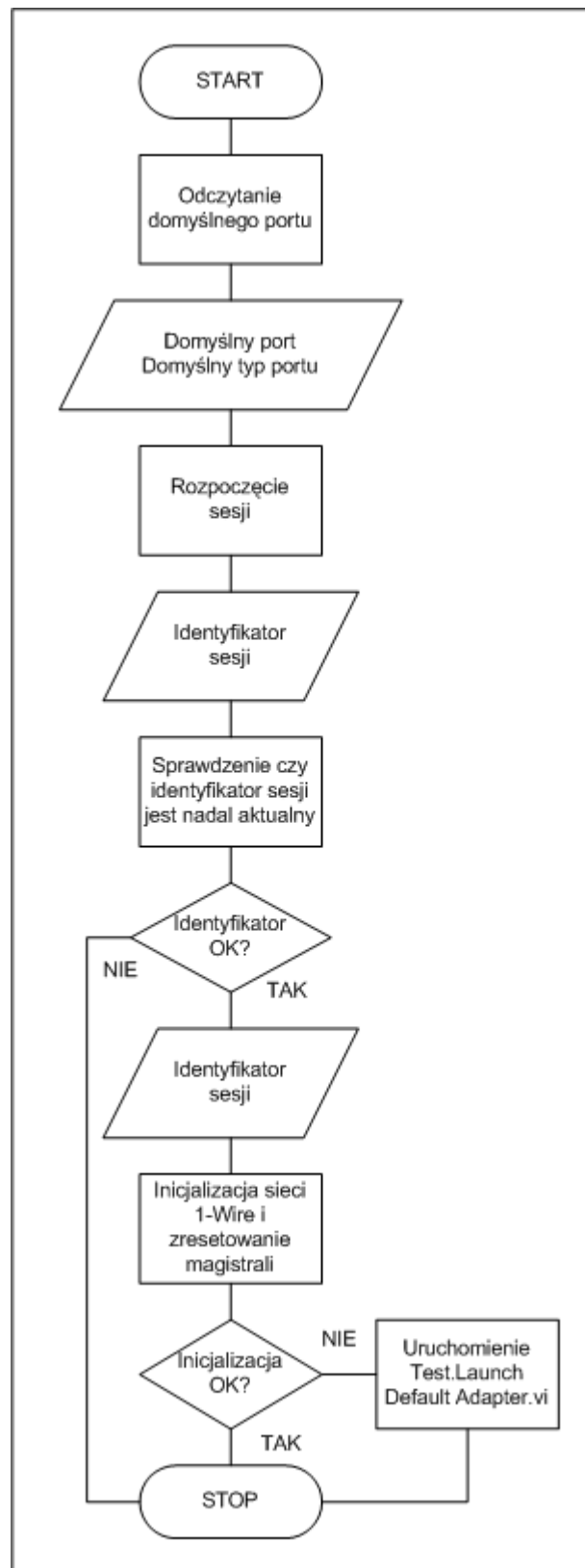
W module pomiarowym, zgodnie ze schematem na rysunku 6.34 wykonują się następujące operacje:

- inicjalizacja sesji 1-Wire,
- wyszukiwanie czujników,
- akwizycja pomiarów,
- zakończenie sesji.

Schemat blokowy procedury inicjalizacji przedstawiony jest na rysunku 6.36. Program sprawdza w rejestrze systemu Windows, czy zapisana jest informacja o numerze i typie portu, do którego podłączone są układy 1-Wire. Jeżeli taki wpis nie istnieje, program uruchamia subVI *test.Launch Default Adapter.vi*, która wyszukuje kontroler 1-Wire wśród portów COM, LPT, USB (rys. 6.35). W wyniku działania programu inicjalizującego zwracany jest unikalny identyfikator sesji. W środowisku, w którym współistnieje kilka sieci 1-Wire, unikalny identyfikator sesji pozwala na niezależną ich pracę. Dany identyfikator używany i przekazywany jest do pozostałych programów. Po użyciu identyfikator zwalniany jest przez program kończący sesję.



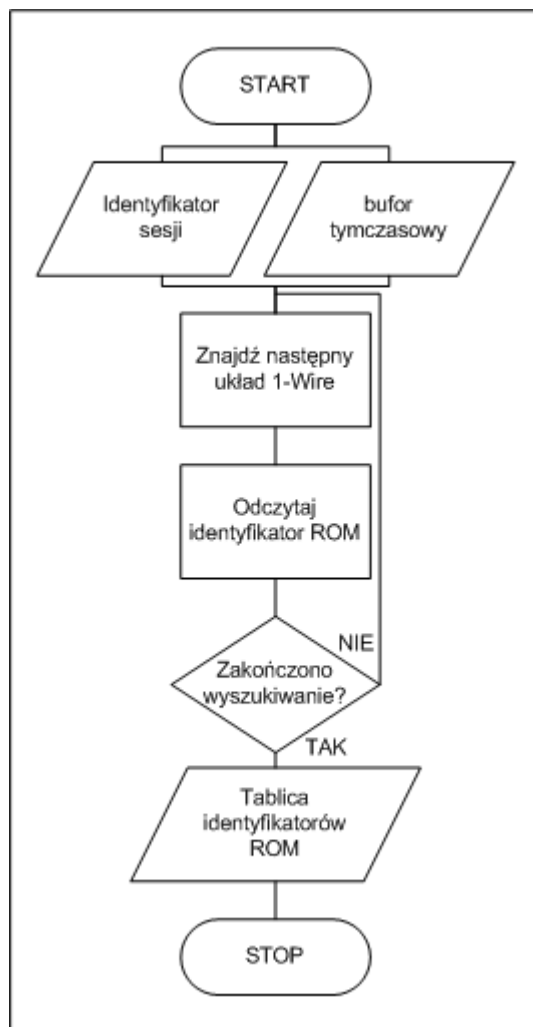
Rys. 6.35 Wyszukiwanie portu kontrolera 1-Wire



Rys. 6.36 Inicjalizacja sesji 1-Wire

Na rysunku 6.37 przedstawiono schemat blokowy programu wyszukującego czujniki podłączone do magistrali 1-Wire. Program wykorzystuje metodę *TMNext*, która przy każdym wywołaniu umieszcza w buforze tymczasowym identyfikatory ROM znalezionych układów.

Po znalezieniu ostatniego urządzenia 1-Wire, kolejne wywołanie funkcji *TMNext* zwraca wartość 0. Wyszukiwanie poszczególnych układów wykonuje się w pętli. W każdej iteracji znaleziony identyfikator ROM zapisywany jest do tablicy, która zwracana jest w wyniku działania programu.



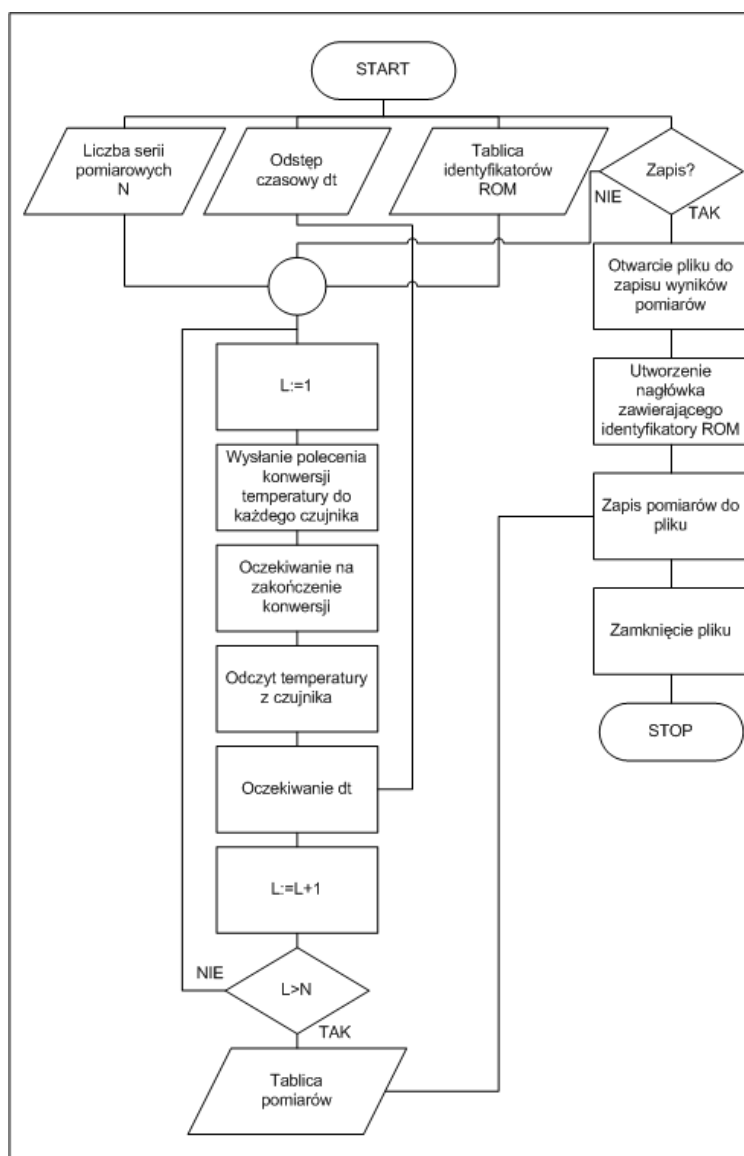
Rys. 6.37 Wyszukiwanie układów w sieci 1-Wire

Po znalezieniu czujników temperatury, ustawieniu parametrów czyli ilości serii pomiarowych i odstępu między nimi, następuje uruchomienie podprogramu wykonującego pomiary temperatury. Zgodnie z wymogami protokołu 1-Wire w celu dokonania pomiaru, należy zachować odpowiednią sekwencję operacji:

- inicjalizacja – impuls *RESET* i *PRESENCE*,
- komenda ROM – polecenie konwersji temperatury,
- oczekiwanie – w zależności o wybranej rozdzielczości czas konwersji jest różny,
- komenda funkcyjna – odczyt obszaru roboczego.

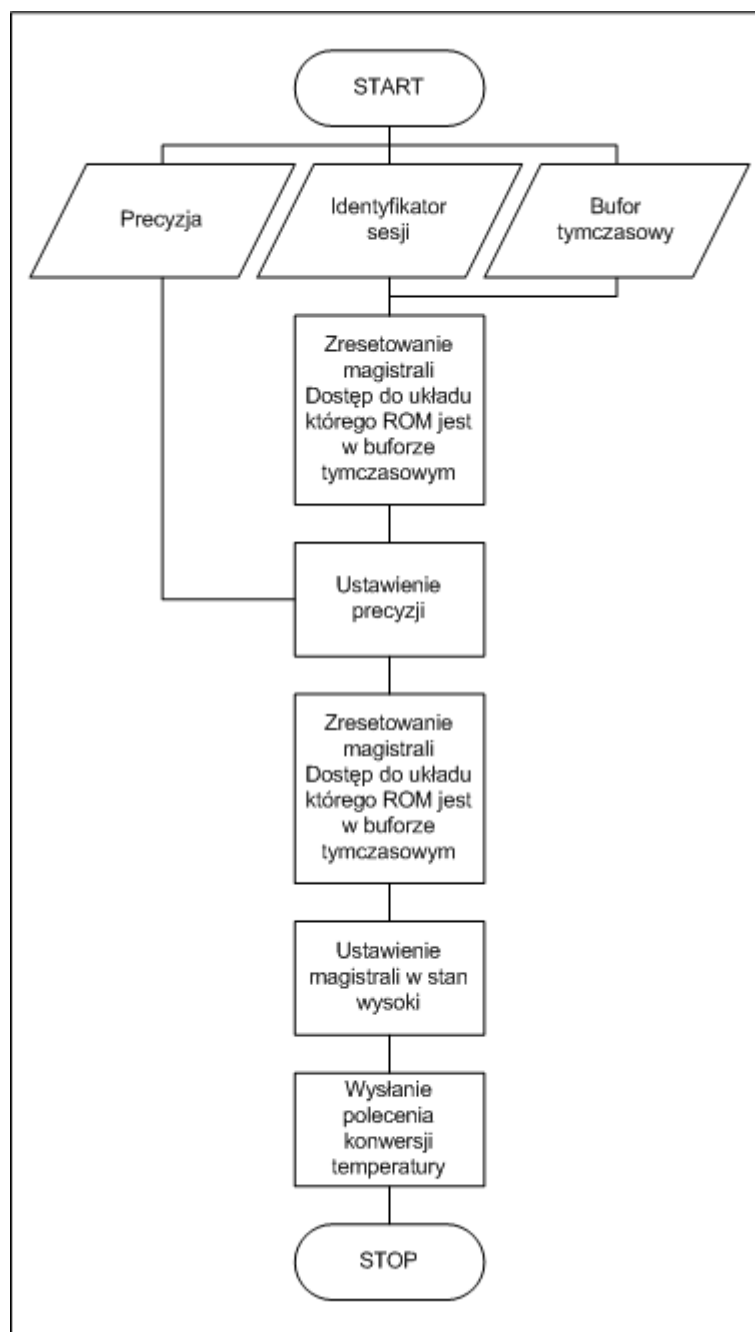
W niniejszej pracy, w celu minimalizacji czasu oczekiwania na kolejne pomiary, wprowadzono sekwencję szeregowo-równoległą. W odróżnieniu od szeregowej, najpierw

wysyła się polecenie konwersji do wszystkich czujników po kolei a następnie, jeżeli jest to konieczne, oczekiwanie na zakończenie konwersji. Dalej w tej samej kolejności, odczytuje się zawartość obszaru roboczego czujników i wyznacza wartość temperatury (w rejestrach czujnika zapisywana jest liczba, którą następnie trzeba przeliczyć na wartość w stopniach Celsjusza lub Farenheita). Wyznaczony doświadczalnie czas wysłania polecenia konwersji jest równy 60 ms, odczytu natomiast 30 ms. Czas oczekiwania na zakończenie konwersji jest równy 100, 200, 400, 800 odpowiednio dla wybranej rozdzielczości 9, 10, 11 lub 12 bitowej. W najgorszym przypadku (rozdzielczości 12 bitowej), odczyt temperatury minimum 14 czujników może rozpocząć się zaraz po wysłaniu polecenia konwersji, ponieważ w tym czasie w pierwszym czujniku zakończy się już konwersja ($14 \cdot 60 = 840$). Wyniki pomiarów zapisywane są do pliku. Schemat blokowy procedury akwizycji temperatury przedstawia rysunek 6.38.

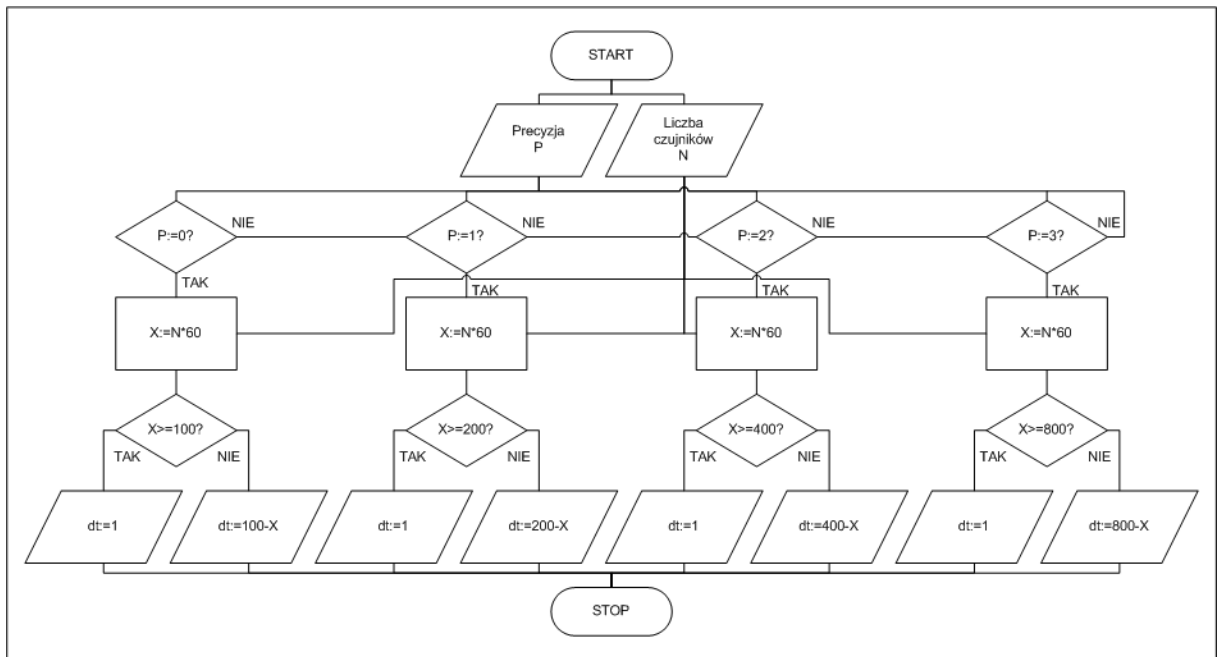


Rys. 6.38 Schemat akwizycji temperatury

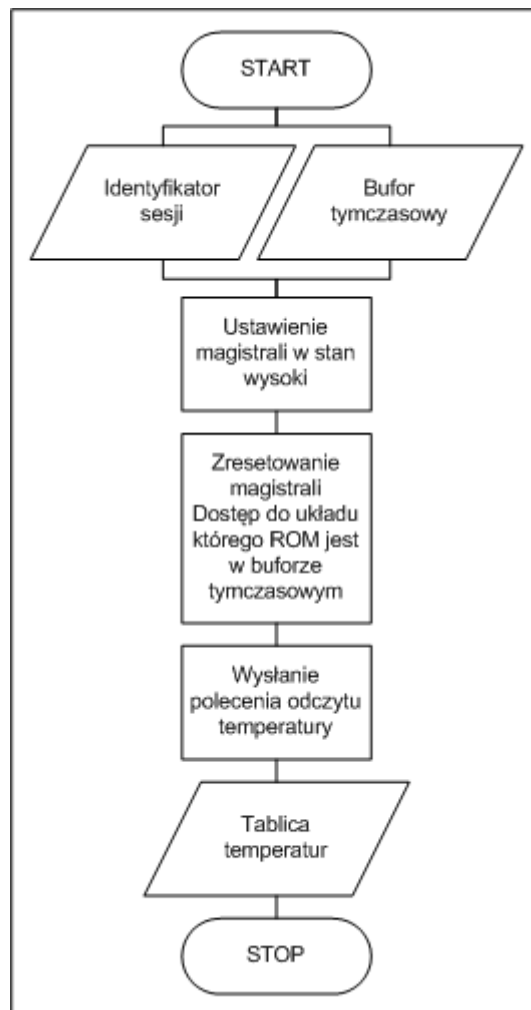
Na rysunkach 6.39÷6.41 przedstawiono szczegółowe schematy blokowe etapów konwersji, oczekiwania i odczytu temperatury czujników DS18B20.



Rys. 6.39 Schemat procedury konwersji temperatury

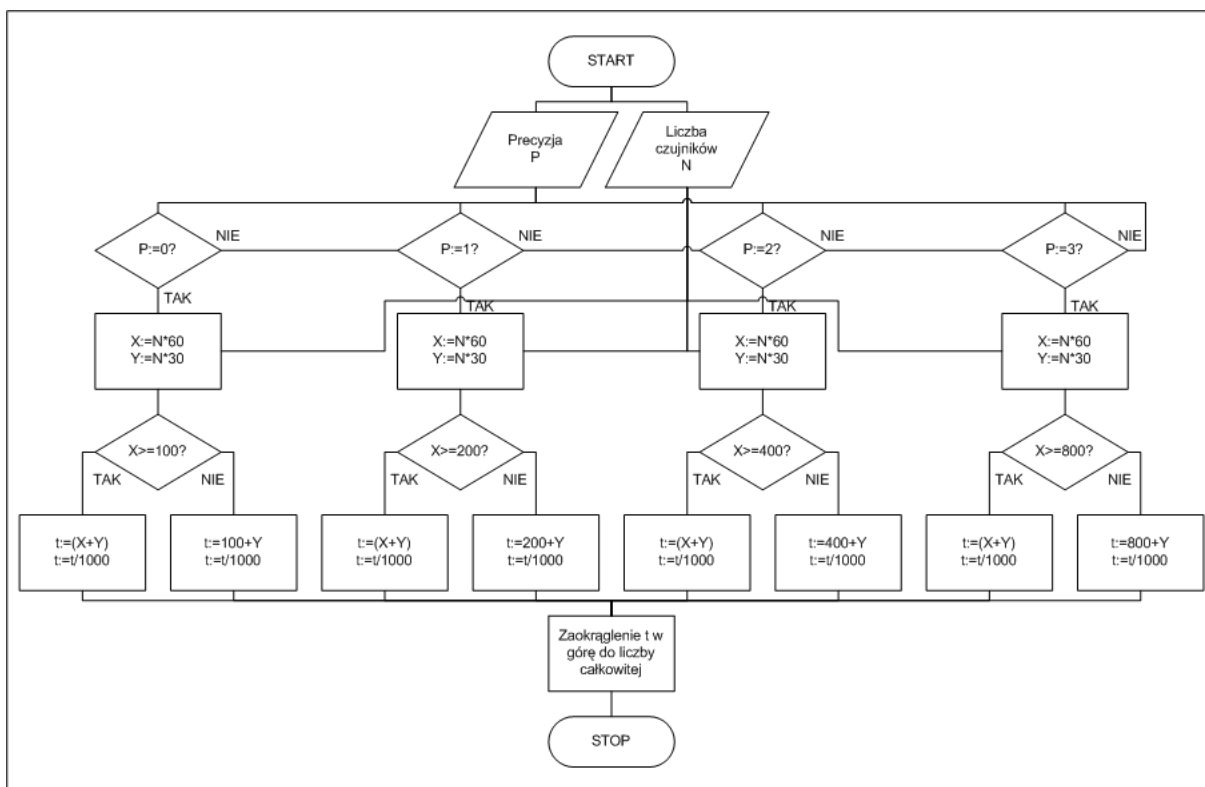


Rys. 6.40 Algorytm wyznaczania czasu potrzebnego do zakończenia konwersji



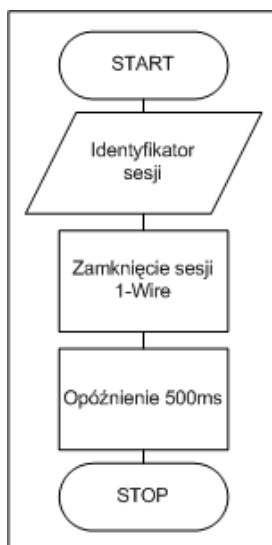
Rys. 6.41 Schemat procedury odczytu temperatury z czujników

Jednym z parametrów programu wykonującego pomiary temperatury jest czas wyrażony w sekundach, określający odstęp między kolejnymi seriami pomiarowymi. W celu zachowania wymagań protokołu 1-Wire, konieczne jest wyznaczenie minimalnej wartości tego czasu, dla danej liczby czujników i wybranej precyzji. W tym celu napisany został odpowiedni subVI. Algorytm programu przedstawiono na rysunku 6.42.



Rys. 6.42 Minimalny odstęp czasowy między kolejnymi seriami pomiarowymi

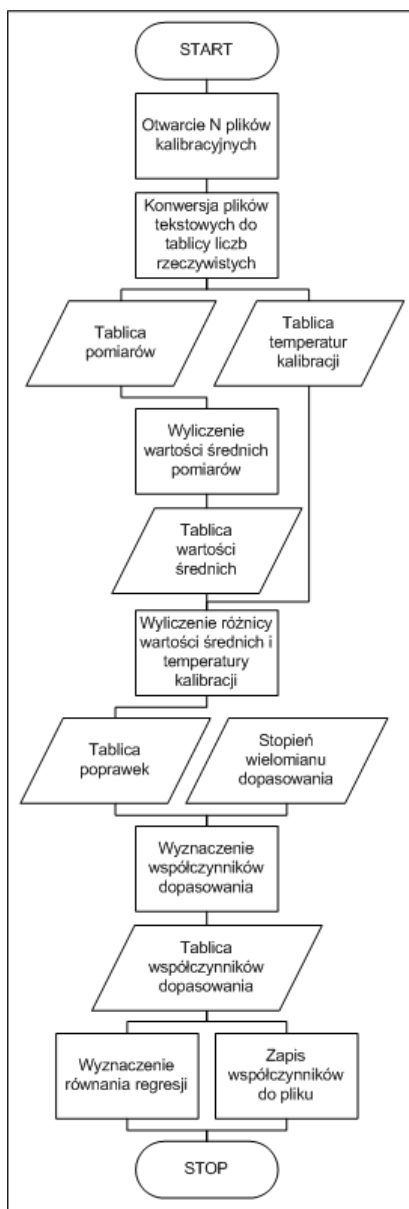
Po zakończeniu wszystkich pomiarów należy zakończyć aktywną sesję 1-Wire i zwolnić przydzielony identyfikator czasowy. Schemat programu przedstawia rysunek 6.43.



Rys. 6.43 Zakończenie sesji 1-Wire

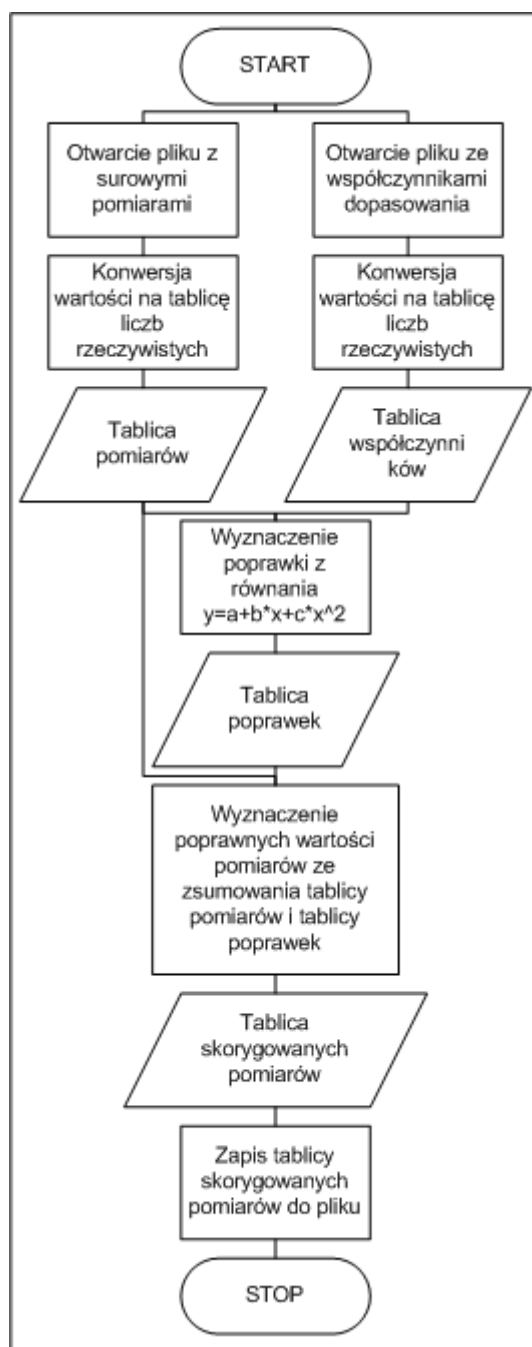
Moduł kalibracyjny, podobnie jak moduł akwizycji temperatury, zajmuje się wykonywaniem pomiarów. Z uwagi na takie samo działanie szczegółowy opis (schematy blokowe, algorytmy) zostanie pominięty. Różnica między aplikacjami polega na wykonywaniu 100 serii pomiarowych, z rozdzielczością 12 bitów bez dodatkowego (określanego przez użytkownika) odstępu czasowego między seriami. Parametrem określanym przez użytkownika jest temperatura, w której wykonywana jest kalibracja. Pozostałe (podane powyżej) są zapisane wewnątrz programu i użytkownik nie może ich zmienić. W wyniku działania programu, na komputerze tworzony jest plik tekstowy, zawierający wyniki pomiarów.

Pomiary kalibracyjne służą do wyznaczania współczynników równania regresji w module dopasowania. Schemat blokowy programu przedstawiono na rysunku 6.44.



Rys. 6.44 Schemat programu wyznaczającego współczynniki dopasowania

Moduł korekty wyznacza rozwiązania równania regresji, na podstawie surowych pomiarów i wyznaczonych współczynników dopasowania. Schemat blokowy programu przedstawia rysunek 6.45. Skorygowane pomiary zapisywane są do pliku tekstowego.



Rys. 6.45 Schemat algorytmu korekty pomiarów

W załączniku 1 do niniejszej pracy znajduje się instrukcja instalacji i użytkowania programu PDKD.

6.6. Pomiary i kalibracja

Podstawową funkcją programu PKDK jest wykonywanie pomiarów temperatury. Pozostałe moduły, operują na wynikach tych pomiarów i pozwalają zwiększyć ich dokładność. W tym celu wykonane zostały pomiary kalibracyjne. Wykorzystano w nich ultratermostat LAUDA UB30 (w punktach 30, 35, 40, 45, 50 °C) i termos z lodem i wodą (w punkcie 0 °C). Wyniki pomiarów pozwoliły wyznaczyć współczynniki krzywej dopasowania poszczególnych czujników. Następnie, za pomocą współczynników skorygowano „surowe” pomiary temperatury. Ultratermostat LAUDA UB30 pozwala zadać temperaturę z dokładnością do setnej części °C. Urządzenie składa się ze sterownika i komory. W komorze znajduje się zanurzony w wodzie cylinder, przez który (poprzez znajdujący się przy dnie otwór) przepływa woda. Cyrkulacja cieczy w komorze sterowana jest przez pompę. Woda przepływająca przez wymiennik ciepła, jest ogrzewana do wartości zadanej przez sterownik. Rysunek 6.46 przedstawia wygląd stanowiska, na którym przeprowadzono pomiary kalibracyjne.



Rys. 6.46 Stanowisko pomiarów kalibracyjnych: a) ultratermostat, b) sterownik c) cylinder z umieszczonymi czujnikami temperatury d) komputer z uruchomioną aplikacją

Do pomiarów kalibracyjnych wykorzystano również punkt zamarzania wody. W tym celu czujniki temperatury umieszczono w termosie, który następnie uzupełniony został lodem i wodą. Do kontroli temperatury w termosie wykorzystano termometr rtęciowy. Na rysunku 6.47 przedstawiono termos z umieszczonymi w nim czujnikami DS18B20.



Rys. 6.47 Zanurzone w termosie czujniki temperatury

Wyznaczone współczynniki dopasowania przedstawiono w tabeli 6.1.

Tab. 6.1 Współczynniki dopasowania czujników temperatury

Stopień wielomianu dopasowania	Współczynniki dopasowania	
	Czujnik 1	Czujnik 2
0	0,0444	0,126483
1	-0,38808	-0,196445
	0,012974	0,009688
2	-0,424857	-0,229903
	0,021516	0,017459
	-0,00018	-0,000164

Do poprawienia „surowych” pomiarów wykorzystano procedurę korekty, dostępną w programie PKDK. Wartości średnie skorygowanych pomiarowych znajdują się w 2 i 3 kolumnie tablicy 6.2. W kolejnych wierszach znajdują się zarówno wartości zależne od wybranego stopnia wielomianu jak i „surowe” pomiary. W tabeli umieszczono również różnice między temperaturą kalibracji i wartością skorygowaną (4 i 5 kolumna). Poszczególne wiersze zawierają wartości dla sześciu punktów, w których wykonywana była kalibracja.

Tab. 6.2 Tabela pomiarów kalibracyjnych i wartości korekt

T_{CAL} i stopień dopasowania	T_{SR} czujnika 1	T_{SR} czujnika 2	$T_{CAL} - T_{SR}$ czujnika 1	$T_{CAL} - T_{SR}$ czujnika 2
0 °C (surowe)	0,424	0,227	0,424	0,227
0 °C (st. 0)	0,380	0,101	0,380	0,101
0 °C (st. 1)	0,041	0,033	0,041	0,033
0 °C (st. 2)	0,008	0,001	0,008	0,001
30 °C (surowe)	29,933	29,862	0,067	0,138
30 °C (st. 0)	29,978	29,989	0,022	0,011

30 °C (st. 1)	29,934	29,955	0,066	0,045
30 °C (st. 2)	29,991	30,008	0,009	0,008
35 °C (surowe)	34,939	34,857	0,061	0,143
35 °C (st. 0)	34,983	34,984	0,017	0,016
35 °C (st. 1)	35,004	34,999	0,004	0,001
35 °C (st. 2)	35,046	35,037	0,046	0,037
40 °C (surowe)	39,806	39,742	0,194	0,258
40 °C (st. 0)	39,850	39,869	0,150	0,131
40 °C (st. 1)	39,934	39,931	0,066	0,069
40 °C (st. 2)	39,952	39,947	0,048	0,053
45 °C (surowe)	44,820	44,757	0,180	0,243
45 °C (st. 0)	44,864	44,884	0,136	0,116
45 °C (st. 1)	45,013	44,995	0,013	0,005
45 °C (st. 2)	44,997	44,980	0,003	0,020
50 °C (surowe)	49,812	49,794	0,188	0,206
50 °C (st. 0)	49,856	49,921	0,144	0,079
50 °C (st. 1)	50,070	50,080	0,070	0,080
50 °C (st. 2)	50,012	50,027	0,012	0,027

Do oceny skuteczności korekty w zależności od wybranego stopnia wielomianu dopasowania, wyznaczono współczynnik c : średni błąd kwadratowy w punktach kalibracji. Równanie (1) pokazuje sposób jego wyliczenia.

$$c = \frac{1}{n} \cdot \sqrt{(T_{CAL1} - T_{SR1})^2 + (T_{CAL2} - T_{SR2})^2 + (...) + (T_{CALn} - T_{SRn})^2} \quad (1)$$

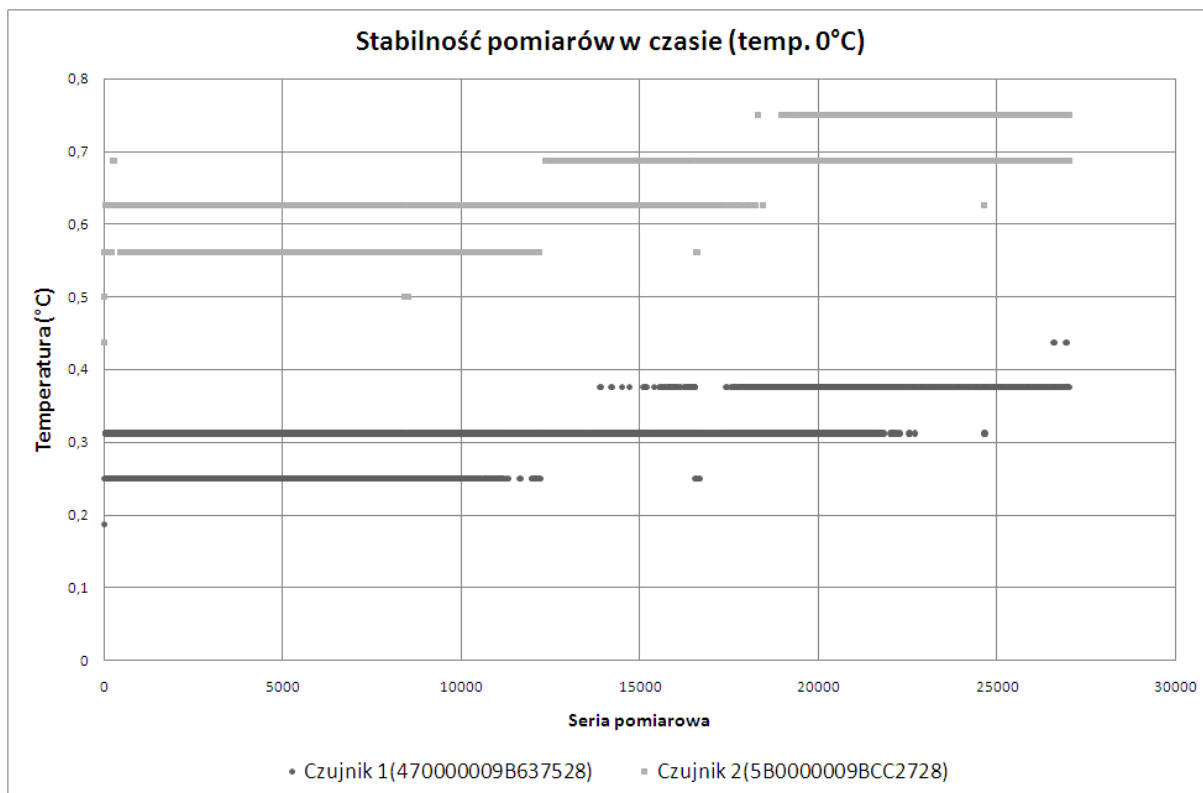
Wartości współczynnika c zestawiono w tablicy 6.3. Dzięki kalibracji czujników DS18B20 udało się osiągnąć 10 krotnie większą, niż fabryczna, dokładność pomiarów.

Tab. 6.3 Tablica współczynników kwadratowych

Stopień wielomianu dopasowania	Średni błąd kwadratowy w punktach kalibracji	
	Czujnik 1	Czujnik 2
0	0,042	0,032
1	0,02	0,019
2	0,011	0,012

W pracy wykonano też serię pomiarów trwającą 6 godzin. Badanie miało na celu pokazanie stabilności czujników w czasie. Pomiary zostały wykonane z użyciem termosu z lodem i wodą. W celu zapewnienia stałej temperatury, w trakcie pomiarów uzupełniano termos lodem, przez co w pomiarach występowały skoki temperatury. Na rysunku 6.48 przedstawiono wykres pomiarów stabilności. Widoczne na wykresie skoki temperatury wynikają również z dyskretyzacji pomiarów. Czujniki DS18B20 są czujnikami cyfrowymi, dysponującymi określoną rozdzielczością (regulowaną w zakresie 9÷12 bitów) i zmiany w

bitach powodują zmianę temperatury. Przedstawione na wykresie wartości są wynikiem „surowych” nieskalibrowanych pomiarów.



Rys. 6.48 Stabilność pomiarów w czasie

7. Podsumowanie i wnioski

1-Wire jest ciekawą technologią komunikacji wykorzystywaną w technice pomiarowej. Zaprojektowana do zastosowań w motoryzacji (jako iButton) została doceniona za prostotę (jeden przewód) i łatwą rozbudowę. Obecnie w oparciu o tę technologię buduje się sieci pomiarowo-kontrolne oraz systemy autoryzacji dostępu. Koszty układów w porównaniu do innych produktów są niższe. Czujnik DS18B20 kosztuje 4 złote, gdzie cena popularnego czujnika PT-100 o zbliżonych parametrach wynosi 50 zł.

Niniejsza praca składa się z dwóch części: teoretycznej i praktycznej. Część teoretyczna poświęcona jest zaznajomieniu się z tematyką inteligentnych budynków i wykorzystywanych w nich rozwiązaniach technicznych. Praca pozwala zapoznać się z metodami komunikacji stosowanymi w instalacjach inteligentnych budynków i wykorzystywanymi protokołami komunikacyjnymi. W dalszej części przedstawiono technologię 1-Wire. Opisano działanie układów, sposoby zasilania, wymagania dotyczące sygnałów czasowych, sposób wykorzystania unikalnego identyfikatora ROM. Przedstawiono problematykę obsługi błędów transmisji wykorzystującą sumy kontrolne CRC. Pokazano metodę zwiększenia dokładności

pomiarów poprzez kalibrację oraz analizę niepewności. Przedstawiono protokół 1-Wire: jego warstwy, rodzaje medium, prędkość działania, topologię połączeń i zasięg sieci. Zaprezentowano rodzaje i parametry układów pracujących w technologii 1-Wire. W celu spełnienia wymagań związanych z przeznaczeniem zbudowanego układu pomiarowego, dokonano wyciągu z dokumentacji technicznej urządzeń i wybrano odpowiednie podzespoły.

W części praktycznej pracy wykonano układ pomiarowy, składający się z kontrolera „USB 1-Wire” opartego na układzie DS2490 (MP00200) i pary czujników DS18B20 o regulowanej rozdzielczości. Dla sprawdzenia użyteczności czujników w środowisku LabVIEW napisano aplikację. Posłużyła ona do mierzenia temperatury, kalibracji czujników i korekty wykonanych pomiarów.

Wybór graficznego środowiska programowania umożliwił łatwą budowę i modyfikację aplikacji. Programowanie graficzne pokazuje kod w sposób bardziej przejrzysty i łatwiejszy do zrozumienia. Elastyczność LabVIEW pozwala zmieniać kod z pomocą kilku kliknięć myszką, zamiast przeszukiwania i poprawiania wielu linii tradycyjnego kodu. Środowisko to posiada rozbudowane narzędzia debugowania. Bardzo przydatną funkcją jest podgląd wykonywanych aktualnie działań w uruchomionej aplikacji. Wykorzystanie techniki podprogramów subVI umożliwia łatwą rozbudowę programu. Ta sama procedura komunikacji z czujnikami została użyta łącznie w 3 modułach aplikacji. W module pomiarowym, kalibracyjnym i podglądu aktualnej temperatury. Uważam, że programowanie graficzne jest łatwo przyswajalne, co ułatwia pracę początkującym programistom. Często grafika potrafi powiedzieć więcej niż skomplikowana składnia instrukcji. Programowanie w LabVIEW jest spójne. Dobrym zwyczajem jest tworzenie programu tak, aby zajmował obszar nie większy niż rozmiar ekranu. Dzięki czemu inni użytkownicy aplikacji w krótszym czasie poznają zasadę jego działania. Mimo, że cena podstawowej wersji LabVIEW (Basic) środowiska kosztuje około 3 tysięcy złotych, a wersji pełnej (Full) niecałe 7 tys., to jednak programista dostaje narzędzie o ogromnych możliwościach.

W dostępnych aplikacjach komunikacja z czujnikami serii DS18x20 odbywa się w sposób szeregowy. Sekwencję rozpoczyna wysłanie polecenia konwersji temperatury, następnie program oczekuje na zakończenie konwersji w czujniku (czas zależny od wybranej rozdzielczości czujnika, dla 12 bitów jest to 850 ms), a na koniec dane z czujnika są odczytywane. W niniejszej pracy napisany program wykorzystuje inną metodę. Przyjęta szeregowo-równoległa komunikacja z czujnikami polega w pierwszym kroku na przesłaniu polecenia konwersji do wszystkich czujników po kolei. Jeśli to konieczne program oczekuje na zakończenie konwersji w czujnikach i następnie w tej samej kolejności odczytuje dane z

czujników. Zmierzony czas potrzebny na wykonanie 20 serii pomiarowych przy użyciu 5 czujników DS18S20 wyniósł 82 sekundy przy zastosowaniu metody szeregowej. Program korzystający ze zmodyfikowanej metody wykonał tę operację w czasie 23 sekund, co stanowi 3,5-krotne skrócenie czasu pomiaru.

W obecnej formie program współpracuje z czujnikami temperatury DS18B20. Modyfikacja podprogramu odpowiedzialnego za komunikację z czujnikami, umożliwi przesył danych z innymi układami 1-Wire oraz z układami akwizycji, które nie są wyposażone w interfejs 1-Wire. W przypadku potrzeby rozbudowy sieci pomiarowo-kontrolnej, np. poprzez zwiększenie ilości pracujących czujników lub wydłużenia połączeń, należy wziąć pod uwagę wpływ zmian na zasięg i promień sieci. W takim przypadku zastosować należy inny kontroler, dysponujący większym wzmocnieniem sygnału, bądź podzielić sieć na segmenty (topologia przełączana, rozdział 4.3.4).

Do oceny niepewności pomiarów temperatury czujników DS18B20 przeprowadzono kalibrację i pomiary kontrolne, także długotrwałe w celu wyznaczenia stabilności w czasie. Kalibracja czujników wykonana w ultratermostacie umożliwiła wyznaczenie równań opisujących statyczne charakterystyki czujników, a wykorzystując metody regresji określono współczynniki równań zapewniających najlepsze dopasowanie jednej z trzech postaci krzywej kalibracyjnej. Przeliczając „surowe” pomiary, w oparciu otrzymane charakterystyki korekcyjne, osiągnięto 10-krotnie mniejszą niż fabrycznie gwarantowana niepewność pomiaru temperatury. Dzięki kalibracji osiągnięta niepewność serii pomiarów dochodziła do setnej części stopnia Celsjusza przy kwancie dyskretyzacji czujników równym $0,0625^{\circ}\text{C}$. Badanie stabilności wykazało, warunki środowiskowe nie mają istotnego wpływu na niepewność pomiarów. Obserwowane okresowo występujące zmiany były wynikiem kwantowania.

W pracy osiągnięto założony cel praktyczny w postaci stworzenia aplikacji pozwalających poprzez sieć 1-Wire zbierać pomiary z czujników temperatury. W przyszłości stworzony układ pomiarowy może być rozwijany, np. poprzez dodanie innych rodzajów układów pomiarowych do regulacji mikroklimatu wewnątrz pomieszczeń: czujników wilgotności powietrza lub stężenia CO_2 oraz układów sterujących, kontroli i autoryzacji dostępu.

8. Spis rysunków

Rys. 3.1 Implementacja modelu ISO/OSI w systemach BAS [21]	8
Rys. 3.2 Warstwy protokołu BACnet – porównanie z OSI [21]	11
Rys. 3.3 Schemat urządzenia LonWorks z układem Neuron Chip [21].....	12
Rys. 3.4 Architektura PROFIBUS wobec modelu ISO/OSI [21]	13
Rys. 3.5 Połączenia między urządzeniami, liniami i obszarami w sieci EIB [35]	15
Rys. 3.6 Stos komunikacyjny protokołu MODBUS [24].....	17
Rys. 4.1 Wykorzystanie kondensatora wraz z diodą Shottky’ego do zasilania [23].....	26
Rys. 4.2 Zasilanie z magistrali 1-Wire, wydzielone dwoma przełącznikami [23].....	27
Rys. 4.3 Szczelina czasowa zapisu „1” na magistrali 1-Wire [11].....	28
Rys. 4.4 Szczelina czasowa zapisu „0” na magistrali 1-Wire [11]	28
Rys. 4.5 Szczelina czasowa odczytu danych z magistrali 1-Wire [11]	29
Rys. 4.6 Impuls reset i odpowiadający mu impuls obecności [11]	31
Rys. 4.7 Struktura identyfikatora urządzeń 1-Wire	32
Rys. 4.8 8-bitowy wielomian CRC protokołu 1-Wire [11]	33
Rys. 4.9 16-to bitowy wielomian CRC protokołu 1-Wire [11]	34
Rys. 4.10 Wykres wartości błędów pomiarowych i wyliczonej krzywej [10]	35
Rys. 4.11 Wykres krzywej skompensowanych błędów pomiarowych [10].....	36
Rys. 4.12 Warstwowa architektura protokołu sieciowego 1-Wire	37
Rys. 4.13 Liniowa topologia sieci 1-Wire [9]	41
Rys. 4.14 Odcinkowa topologia sieci 1-Wire [9]	41
Rys. 4.15 Topologia gwiazdy w sieci 1-Wire [9].....	42
Rys. 4.16 Topologia przełączana sieci 1-Wire [9]	42
Rys. 4.17 Topologia rozproszonej impedancji w sieci 1-Wire [9].....	43
Rys. 4.18 Prosty interfejs sieciowy [9].....	44
Rys. 4.19 Interfejs wyposażony w tranzystor i wzmacniacz [9]	45
Rys. 4.20 Interfjes sieciowy z filtrem RC [9].....	45
Rys. 4.21 Zaawansowany interfejs sieciowy [9]	46
Rys. 5.1 Diagram blokowy układu DS2431 - 1Kb EEPROM [15].....	47
Rys. 5.2 Czujnik temperatury 1-Wire DS18B20 [12]	49
Rys. 5.3 Schemat ideowy interfejsu 1-Wire	50
Rys. 5.4 Schemat blokowy układu DS2450 [16]	52
Rys. 5.5 Przełącznik 1-Wire – schemat ideowy	53

Rys. 5.6 Czujnik ciśnienia Telaire 6004 [29]	54
Rys. 5.7 Czujnik wilgotności RHU1015 [26]	55
Rys. 6.1 Program FontEd w trybie edycji	57
Rys. 6.2 Ustawienia matrycy edytowanego zestawu znaków w programie FontEd	58
Rys. 6.3 Podgląd tworzonego znaku w programie FontEd	58
Rys. 6.4 Okno programu Analog Configurator	59
Rys. 6.5 Wybór wskaźnika w programie Lampomittari.....	59
Rys. 6.6 Okno główne programu Thermometer	59
Rys. 6.7 Ustawienia główne	60
Rys. 6.8 Konfiguracja MikroLan w programie Thermometer.....	60
Rys. 6.9 Konfiguracja grup czujników w programie Thermometer.....	61
Rys. 6.10 Ustawienia kolorów w programie Thermometer.....	61
Rys. 6.11 Ustawienia obsługi obrazów w programie Thermometer	62
Rys. 6.12 Konfiguracja sensorów w programie Thermometer.....	63
Rys. 6.13 Ustawienia alarmów czujników w programie Thermometer	63
Rys. 6.14 Wykrywanie trendu zmiany temperatury w programie Thermometer.....	64
Rys. 6.15 Korekcja błędów pomiarów czujników temperatury w programie Thermometer ...	64
Rys. 6.16 Okno programu Thermometer z przykładowymi wskaźnikami.....	65
Rys. 6.17 Przykładowy wskaźnik analogowy programu Thermometer	65
Rys. 6.18 Wyświetlacz LCD prezentujący pomiar z pięciu czujników temperatury	66
Rys. 6.19 Wykres historii czujnika temperatury	66
Rys. 6.20 Wykres trendu wybranego sensora.....	67
Rys. 6.21 Wskaźnik prezentujący pomiar z pięciu czujników temperatury.....	67
Rys. 6.22 Listing pliku utworzonego wskaźnika.....	68
Rys. 6.23 Przykładowy wygląd panelu frontowego	69
Rys. 6.24 Przykładowy wygląd diagramu blokowego	69
Rys. 6.25 Przykład kontrolek umieszczanych na panelu frontowym programu	70
Rys. 6.26 Przykład wskaźników umieszczanych na panelu frontowym programu.....	70
Rys. 6.27 Struktury i pętle środowiska LabVIEW	71
Rys. 6.28 Przykładowe przewody łączące obiekty na diagramie blokowym.....	71
Rys. 6.29 Ikona programu subVI ze zdefiniowanymi połączeniami.....	72
Rys. 6.30 Paleta funkcji środowiska LabVIEW	72
Rys. 6.31 Paleta podstawowych funkcji do budowy aplikacji w LabVIEW.....	73
Rys. 6.32 Układ czujników DS18B20 wykorzystywanych w pracy	74

Rys. 6.33 Moduł MP00200 a) bez pokrywy, b) w zamkniętej obudowie, c) z podłączonymi czujnikami temperatury, d) w połączeniu z komputerem.....	75
Rys. 6.34 Schemat blokowy programu PKDK.vi.....	76
Rys. 6.35 Wyszukiwanie portu kontrolera 1-Wire	77
Rys. 6.36 Inicjalizacja sesji 1-Wire	78
Rys. 6.37 Wyszukiwanie układów w sieci 1-Wire	79
Rys. 6.38 Schemat akwizycji temperatury	80
Rys. 6.39 Schemat procedury konwersji temperatury	81
Rys. 6.40 Algorytm wyznaczania czasu potrzebnego do zakończenia konwersji.....	82
Rys. 6.41 Schemat procedury odczytu temperatury z czujników.....	82
Rys. 6.42 Minimalny odstęp czasowy między kolejnymi seriami pomiarowymi.....	83
Rys. 6.43 Zakończenie sesji 1-Wire	83
Rys. 6.44 Schemat programu wyznaczającego współczynniki dopasowania	84
Rys. 6.45 Schemat algorytmu korekty pomiarów	85
Rys. 6.46 Stanowisko pomiarów kalibracyjnych: a) ultratermostat, b) sterownik c) cylinder z umieszczonymi czujnikami temperatury d) komputer z uruchomioną aplikacją ...	86
Rys. 6.47 Zanurzone w termosie czujniki temperatury	87
Rys. 6.48 Stabilność pomiarów w czasie	89

9. Spis tabel

Tab. 3.1 Usługi używane w komunikacji między urządzeniami w budynku [31]	10
Tab. 3.2 Zależność prędkości transmisji od długości medium transmisyjnego	14
Tab. 3.3 Charakterystyka pracy sieci CANopen [2].....	22
Tab. 4.1 Wartości kompensowanej temperatury i błędów pomiarowych [10].....	36
Tab. 4.2 Komendy ROM warstwy sieci protokołu 1-Wire [11].....	38
Tab. 4.3 Komendy warstwy transportowej protokołu 1-Wire [11]	38
Tab. 4.4 Porównanie czasu trwania impulsów w trybie regularnym i overdrive	40
Tab. 5.1 Układy pamięciowe 1-Wire.....	48
Tab. 5.2 Czujniki temperatury 1-Wire	49
Tab. 5.3 Spis interfejsów 1-Wire wraz z opisem.....	50
Tab. 5.4 Układy RTC pracujące w sieci 1-Wire.....	52
Tab. 5.5 Przełączniki 1-Wire.....	54

Tab. 5.6 Parametry układu Telaar 6004 [29].....	54
Tab. 5.7 Parametry układu RHU1015 [26]	55
Tab. 5.8 Parametry układu iButton DS1923 [13].....	55
Tab. 6.1 Współczynniki dopasowania czujników temperatury	87
Tab. 6.2 Tabela pomiarów kalibracyjnych i wartości korekt	87
Tab. 6.3 Tablica współczynników kwadratowych	88

10. Bibliografia

1. Blim M., 2004, *BMS - integracja systemów i dezintegracja oczekiwań*, Zabezpieczenia, nr 6, strony 30-36,
2. CAN in Automation, 23.12.2003, *Bit timing*, Pobrano 15.10.2007 z lokalizacji CANopen protocol: <http://www.can-cia.org/canopen/protocol/bittiming.html>,
3. CAN in Automation, 14.04.2005, *CAN in Automation - CANopen device and application profiles*, Pobrano 11.10.2007 z lokalizacji Controller Area Network (CAN) - CAN in Automation (CiA): <http://www.can-cia.org/canopen/profile>,
4. CAN in Automation, 02.01.2007, *Controller Area Network (CAN) – protocol*, Pobrano 10.10.2007 z lokalizacji Controller Area Network (CAN) - CAN in Automation (CiA): <http://www.can-cia.org/can/protocol>,
5. CAN in Automation, 23.12.2003, *Network Management (NMT)*, Pobrano 15.10.2007 z lokalizacji CANopen protocol: <http://www.can-cia.org/canopen/protocol/nmt.html>,
6. CAN in Automation, 23.12.2003, *Process Data Object (PDO)*, Pobrano 15.10.2007 z lokalizacji CANopen protocol: <http://www.can-cia.org/canopen/protocol/pdo.html>,
7. CAN in Automation, 23.12.2003, *Service Data Object (SDO)*, Pobrano 15.10.2007 z lokalizacji CANopen protocol: <http://www.can-cia.org/canopen/protocol/sdo.html>,
8. CAN in Automation, 23.12.2003, *Special function objects*, Pobrano 15.10.2007 z lokalizacji CANopen protocol: <http://www.can-cia.org/canopen/protocol/special.html>,
9. Dallas Semiconductor Corp., 2001, *Application Note 148 Guidelines for Reliable 1-Wire Networks*, [s.l.],
10. Dallas Semiconductor Corp., 2002, *Application Note 208 Curve Fitting the Error of a Bandgap-Based Digital Temperature Sensor*, [s.l.],
11. Dallas Semiconductor Corp., 1997, *Book of iButton Standards*, [s.l.],

12. Dallas Semiconductor Corp., 2007, *DS18B20 Programmable Resolution 1-Wire Digital Thermometer*, [s.l.],
13. Dallas Semiconductor Corp., 2007, *DS1923 Hygrochron Temperature/Humidity Logger iButton with 8kB Data Log Memory*, [s.l.],
14. Dallas Semiconductor Corp., 2004, *DS2408 1-Wire 8-channel Addressable Switch*, [s.l.],
15. Dallas Semiconductor Corp., 2007, *DS2431 1024-bit 1-Wire EEPROM*, [s.l.],
16. Dallas Semiconductor Corp., 2006, *DS2450 1-Wire Quad A/D Converter*, [s.l.],
17. Dallas Semiconductor Corp., 2001, *DS2490 USB to 1-Wire bridge chip*, [s.l.],
18. Doliński J., 2003, *Zimno, zimno, ciepło, ciepło czyli Lampomittari*, Elektronika Praktyczna, nr 6, strony 77-80,
19. Echelon Corporation, 2007, *Neuron Chips*, Pobrano 20.08.2007 z lokalizacji Witryna korporacji Echelon: <http://www.echelon.com/developers/lonworks/neuron.htm>,
20. ISO (International Organization for Standardization), 05.04.2007, *Publicly Available Standards*, Pobrano 30.04.2007 z lokalizacji Witryna organizacji ISO: <http://standards.iso.org/ittf/PubliclyAvailableStandards>,
21. Kabza Z., Kostyrko K., Zator S., Łobzowski A., Szkolnikowski W., 2005, *Regulacja mikroklimatu pomieszczeni*, Agenda Wydawnicza PAK-u, Warszawa,
22. KNX Assosiation, 02.08.2007, *KNX Standard – Introduction*, Pobrano 03.10.2007 z lokalizacji KNX Assosiation - [Offical website]: <http://www.knx.org/knx-standard/introduction>,
23. Maxim Integrated Products, 04.02.2002, *Application Note 147 Supplying Power via the 1-Wire Bus*, [s.l.],
24. Modbus Organization, Inc., 2006, *MODBUS Application Protocol 1.1b*, Modbus Organization, Inc., North Grafton,
25. Petykiewicz P., 2004, *Nowoczesna instalacja elektryczna w inteligentnym budynku: przesłanki, zasady, techniczna realizacja, osprzęt*, Centralny Ośrodek Szkolenia i Wydawnictw SEP, Warszawa,
26. Scimarec, 2000, *Humidity sensors - type HU10 relative humidity transmitter unit*, [s.l.],
27. SMARTech Sp. z o.o., 05.11.2007, *Zasada działania X10*, Pobrano 05.11.2007 z lokalizacji Inteligentny Dom w systemie X10: <http://www.x10.pl/?60331305>,
28. Solnik W., Zajda Z., 2004, *Komputerowe sieci przemysłowe Profibus DP i MPI*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław,
29. Telaire, 2002, *CO2 OEM Module – 6004*, [s.l.],
30. Widorski T., 2005, *Samochodowe sieci informatyczne*, Poradnik serwisowy, nr 5,

31. Wikimedia Foundation, Inc., 02.05.2007, *BACnet*, Pobrano 07.06.2007 z lokalizacji Wikipedia, the free encyclopedia: <http://en.wikipedia.org/wiki/Bacnet>,
32. Wikimedia Foundation, Inc., 09.10.2007, *CANopen*, Pobrano 15.10.2007 z lokalizacji Wikipedia, the free encyclopedia: <http://en.wikipedia.org/wiki/Canopen>,
33. Wikimedia Foundation, Inc., 21.11.2006, *LonTalk*, Pobrano 19.08.2007 z lokalizacji Wikipedia, the free encyclopedia: <http://en.wikipedia.org/wiki/LonTalk>,
34. Wikimedia Foundation, Inc., 08.03.2007, *Profibus*, Pobrano 21.08.2007 z lokalizacji Wikipedia, the free encyclopedia: <http://en.wikipedia.org/wiki/Profibus>,
35. Wikimedia Foundation, Inc., 09.09.2007, *System EIB*, Pobrano 03.10.2007 z lokalizacji Wikipedia, wolna encyklopedia: http://pl.wikipedia.org/wiki/System_EIB,
36. Wikimedia Foundation, Inc., 07.08.2007, *Urządzenie wykonawcze*, Pobrano 21.08.2007 z lokalizacji Wikipedia, wolna encyklopedia: http://pl.wikipedia.org/wiki/Urządzenie_wykonawcze,
37. X10.com, 05.11.2007, *X10 Product Catalog*, Pobrano 05.11.2007 z lokalizacji X10 Homepage: <http://www.x10.com/catalog>,
38. X10.com, 05.11.2007, *X10 Transmission Theory*, Pobrano 05.11.2007 z lokalizacji X10.com: <http://www.x10.com/support/technology1.htm>,

Załącznik 1

Instrukcja instalacji sterowników 1-Wire

1. Pierwszym krokiem jest wybór sterowników 1-Wire ze strony producenta. Pod adresem <http://www.maxim-ic.com/products/ibutton/software/tmex> należy pobrać odpowiednie dla używanego systemu operacyjnego sterowniki. Dostępne są wersje pod system Windows XP i Vista zarówno w wersji 32-bitowej i 64-bitowej. Instrukcja opierać będzie się o sterowniki w wersji 4.00, dla systemu Windows XP x86.
2. Pod adresem http://files.dalsemi.com/auto_id/licensed/install_1_wire_drivers_v400.msi pobrać można wyżej wymienione sterowniki. Instalator oferowany jest w angielskiej wersji językowej.
3. Po pobraniu sterowników należy uruchomić program instalacyjny. W pierwszym kroku instalator informuje o tym, że użytkownik zamierza zainstalować sterowniki do obsługi urządzeń 1-Wire. W celu kontynuowania należy nacisnąć przycisk Dalej (ang. *Next*).



Rysunek 1 Rozpoczęcie instalacji sterowników 1-Wire

4. W kolejnym kroku należy zapoznać się z treścią licencji użytkownika programu. W celu kontynuacji należy wyrazić zgodę zaznaczając opcję ang. *I Agree* a następnie wcisnąć przycisk ang. *Next*.



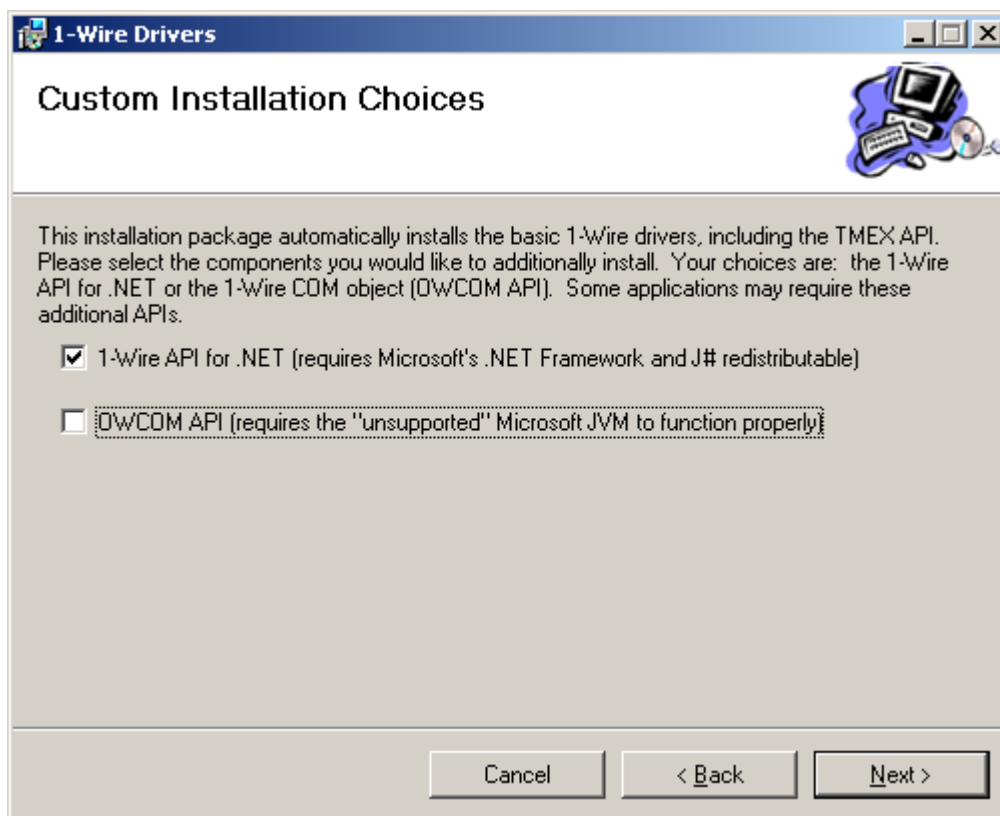
Rysunek 2 Treść licencji użytkownika oprogramowania 1-Wire

5. W kolejnym kroku należy dokonać wyboru instalacji dodatkowych komponentów:

- 1-Wire .net API,
- OWCOM API.

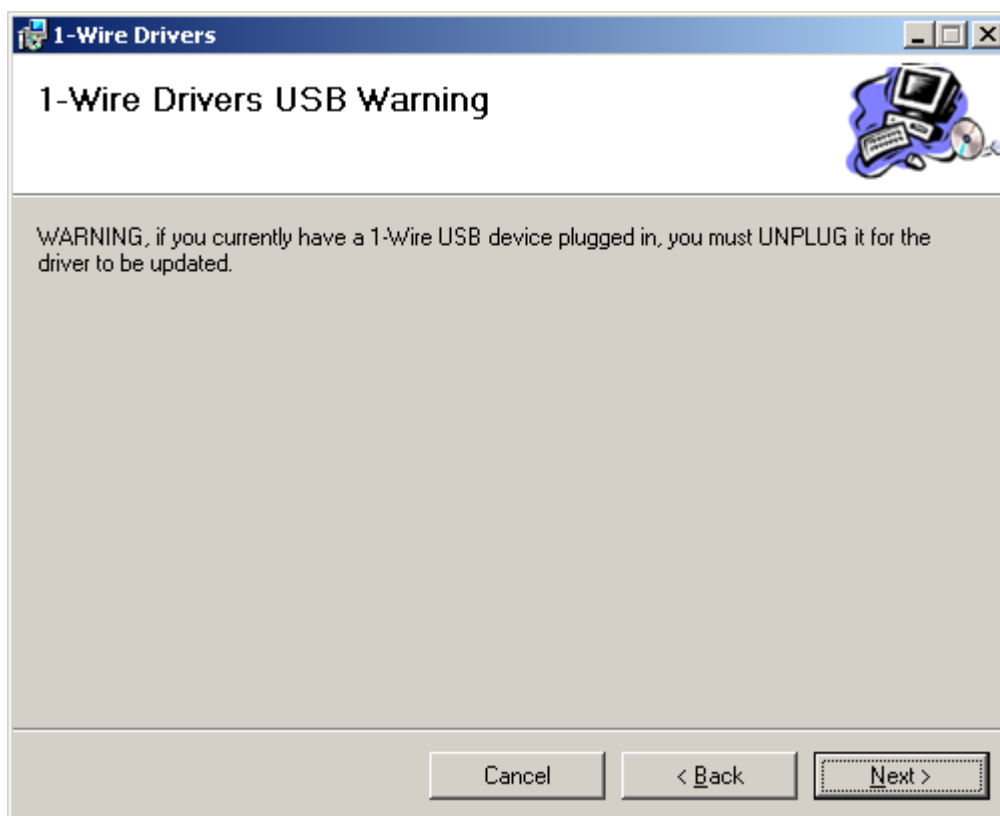
Na swoich stronach internetowych firma Dallas-Maxim ostrzega aby nie instalować komponentu *OWCOM API*, ze względu na brak dostępności technologii wirtualnej maszyny javy (ang. *Microsoft Java Virtual Machine*). Jeżeli w systemie nie ma zainstalowanej technologii MSJVM, podczas instalacji należy odznaczyć komponent *OWCOM API*.

Po wybraniu komponentów, należy przejść do kolejnego kroku wciskając przycisk *Next*.



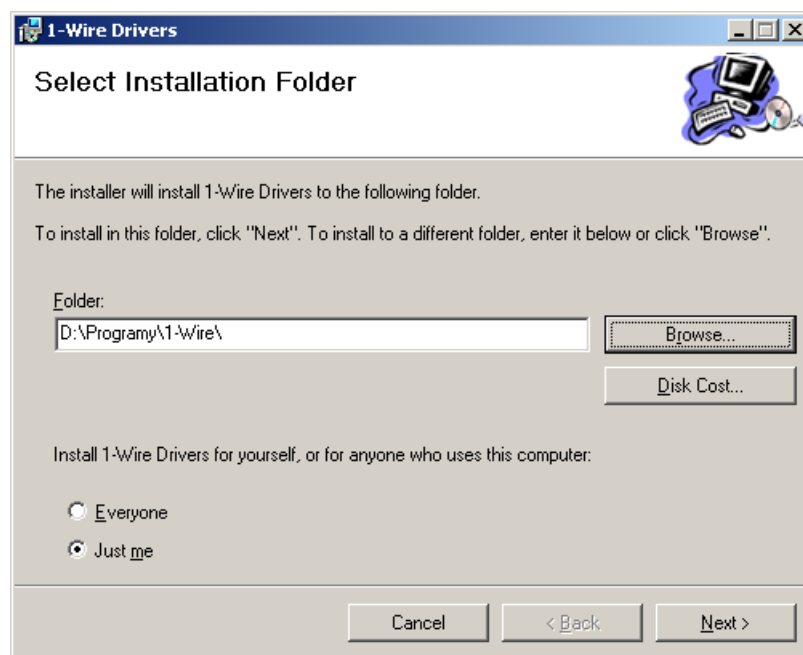
Rysunek 3 Wybór dodatkowych komponentów

6. Instalator wyświetla ostrzeżenie, aby odłączyć od komputera wszystkie urządzenia 1-Wire podłączone do portów USB. W celu kontynuacji należy wcisnąć przycisk *Next*.



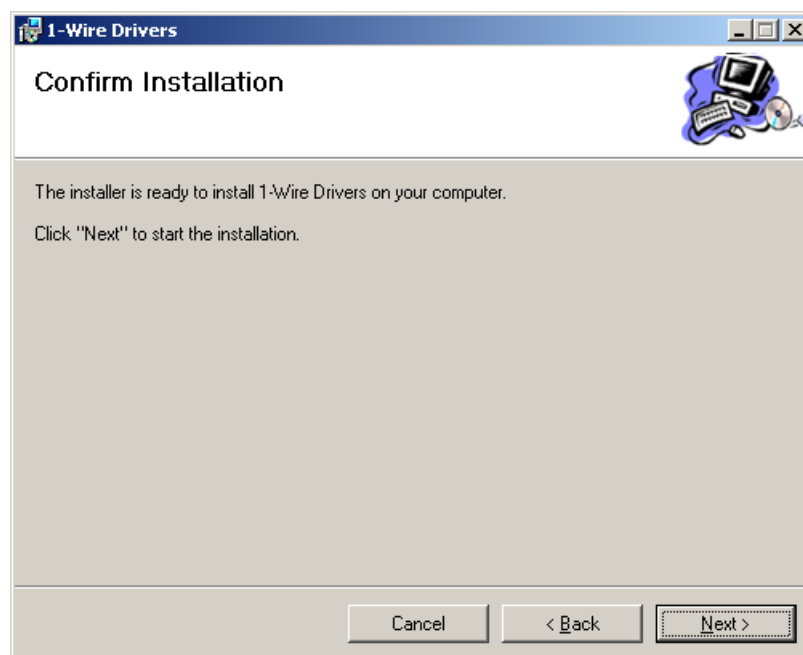
Rysunek 4 Monit o odłączenie urządzeń 1-Wire od portów USB komputera

7. W kolejnym kroku należy wskazać folder docelowy, w którym zostaną zainstalowane sterowniki 1-Wire. Dodatkowo należy wskazać, czy sterowniki mają być zainstalowane dla jednego (obecnie zalogowanego użytkownika) czy dla wszystkich. W celu wyboru ścieżki innej niż domyślnie wskazana przez instalator, należy wcisnąć przycisk przeglądania (ang. *Browse...*) i po wskazaniu ścieżki wcisnąć OK. Po zaakceptowaniu wszystkich ustawień należy wcisnąć przycisk *Next* w celu przejścia do kolejnego kroku.



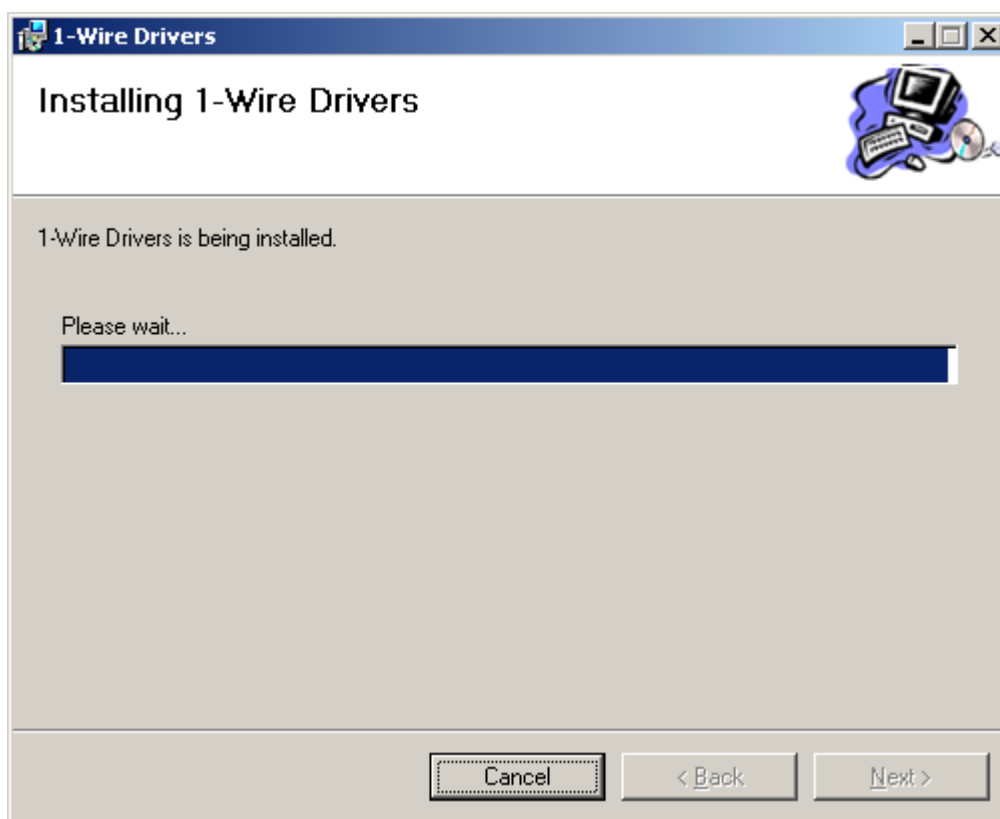
Rysunek 5 Wybór folderu instalacji sterowników

8. W tym miejscu instalator informuje o gotowości do instalacji sterowników. W celu rozpoczęcia instalacji należy wcisnąć przycisk *Next*.



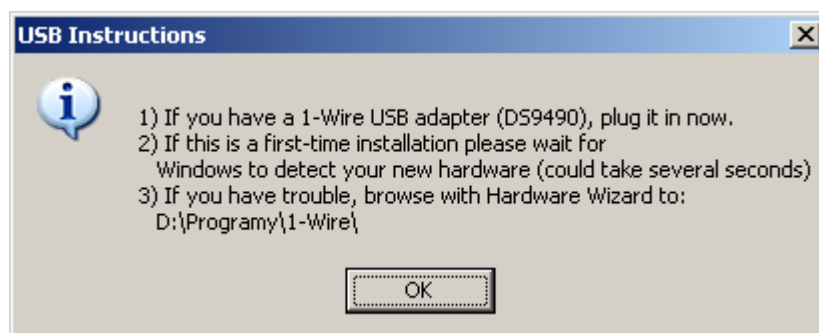
Rysunek 6 Instalator rozpoczyna instalację sterowników

9. W trakcie kopiowania plików na ekranie pasek postępu informuje o etapie instalacji.



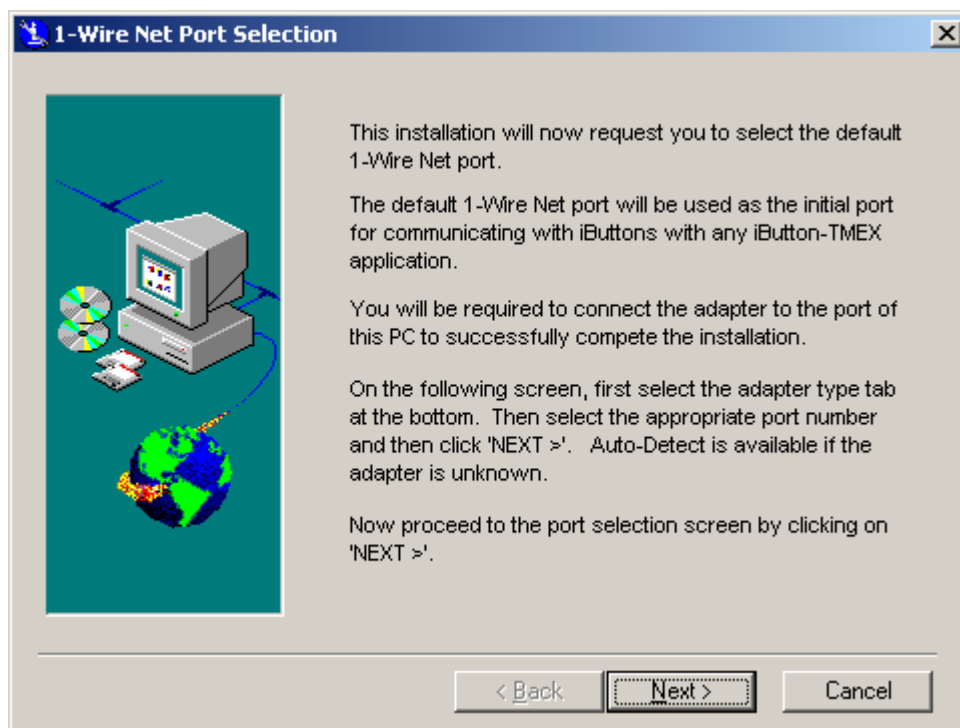
Rysunek 7 Postęp instalacji sterowników

10. W następnym kroku instalator prosi o podłączenie urządzenia (kontrolera) 1-Wire do portu USB. W trakcie pierwszego podłączenia układu, system Windows zainstaluje wymagane sterowniki. Jeżeli system operacyjny ma problem z automatycznym rozpoznaniem położenia sterowników, należy wskazać katalog wybrany w pkt. 7. Po podłączeniu i zainstalowaniu przez system Windows sterowników układu 1-Wire należy wcisnąć przycisk OK.



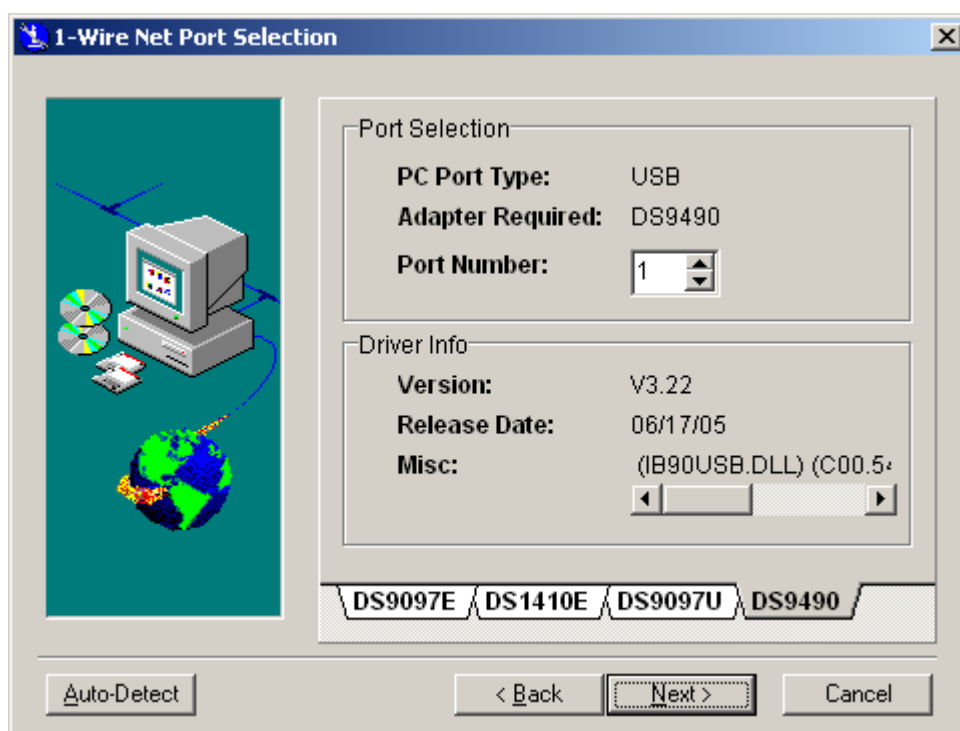
Rysunek 8 Podłączenie kontrolera 1-Wire do portu USB w celu instalacji sprzętu w systemie

11. Instalator sterowników 1-Wire uruchamia program do znajdowania portu, do którego użytkownik podłączył kontroler 1-Wire. W celu rozpoczęcia wyszukiwania należy wcisnąć przycisk *Next*.



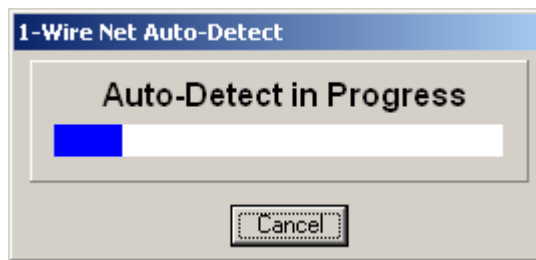
Rysunek 9 Wyszukiwanie portu do którego podłączony jest kontroler 1-Wire

12. W tym kroku należy wskazać odpowiedni port, w zależności o posiadanego kontrolera, lub wcisnąć przycisk automatycznego wyszukiwania (ang. *Auto-Detect*).



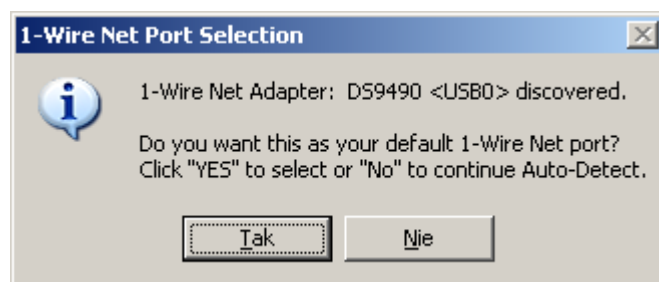
Rysunek 10 Wybór typu kontrolera i portu

13. Postęp wyszukiwania pokazują odpowiednie okno dialogowe.



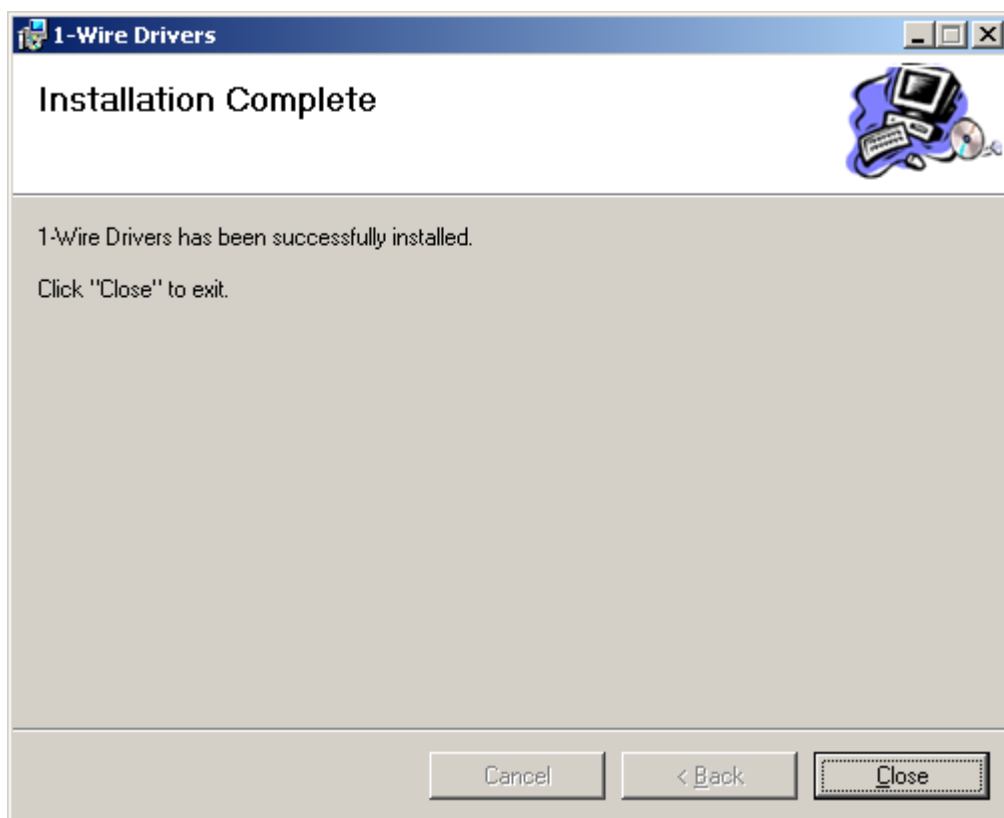
Rysunek 11 Postęp wyszukiwania kontrolera 1-Wire

14. W przypadku znalezienia kontrolera, jego konfigurację można zapisać jako domyślną, wciskając przycisk Tak. W tym przypadku informacje zostaną zapisane do rejestru systemowego. W przeciwnym wypadku (przycisk Nie) nastąpi kontynuacja wyszukiwania.



Rysunek 12 Zapis konfiguracji jako domyślnej

15. Po zatwierdzeniu znalezionej kontrolera, kończy się instalacja sterowników 1-Wire. W tym momencie należy wcisnąć przycisk *Close*.

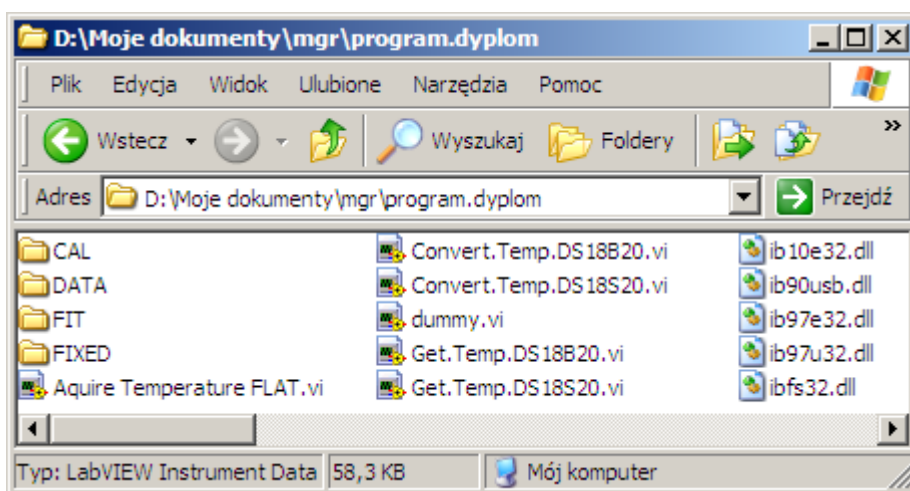


Rysunek 13 Zakończenie instalacji sterowników

Instrukcja użytkownika programu PKDK

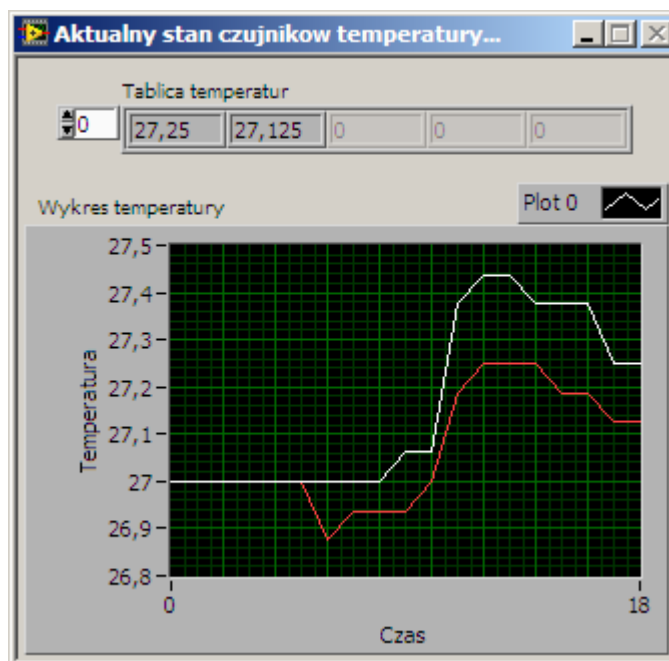
1. W celu wykonywania pomiarów, kalibracji, dopasowania i korekty należy uruchomić program *PKDK.vi* z katalogu programu. Oprócz programu głównego i podprogramów, w katalogu głównym znajdują się podkatalogi zawierające pliki z wynikami przeprowadzonych w programie operacji. W kolejności alfabetycznej występują następujące podkatalogi:

- CAL – pliki z pomiarami kalibracyjnymi,
- DATA – pliki z pomiarami „surowymi”,
- FIT – pliki ze współczynnikami dopasowania,
- FIXED – pliki z poprawionymi pomiarami.



Rysunek 14 Struktura katalogów i plików programu PKDK

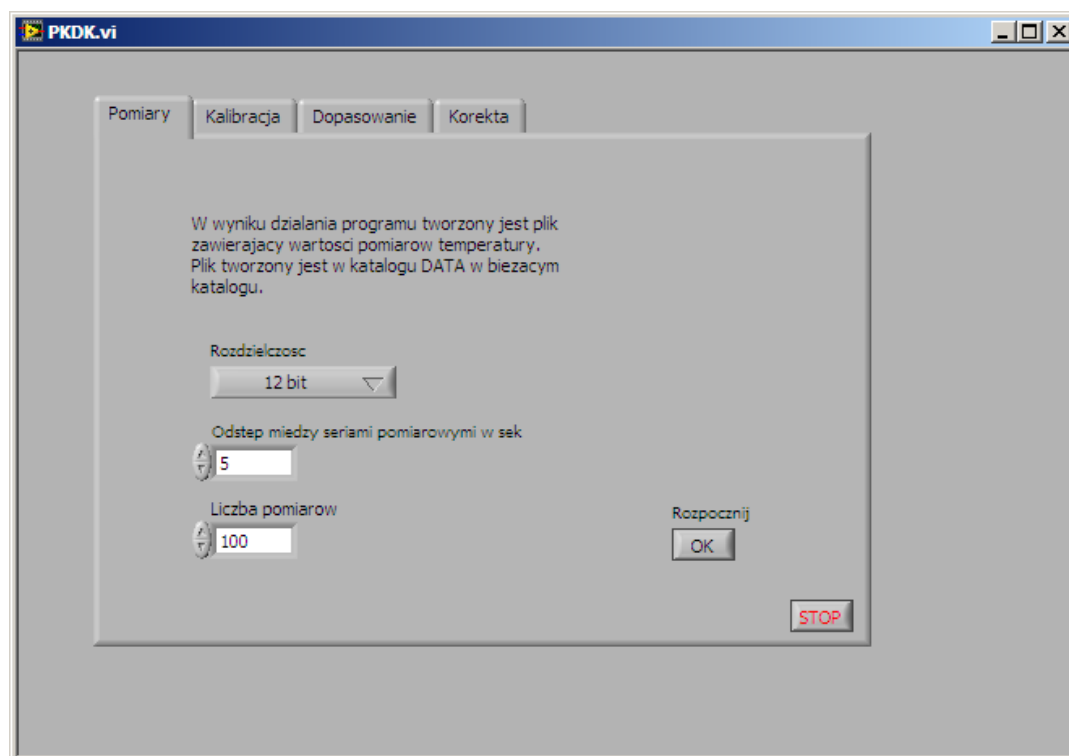
2. Po uruchomieniu programu na ekranie komputera widoczne są dwa okna:
 - PKDK – uruchamianie pomiarów, kalibracji itd.,
 - Aktualny stan czujników – wykres i tablica z aktualnymi wartościami temperatury.



Rysunek 15 Okno programu wskazującego aktualną temperaturę

3. W oknie programu PKDK widoczne są cztery zakładki oddzielające funkcje programu. Zakładka *Pomiary* uruchamia wykonywanie pomiarów z ustawionych parametrów:

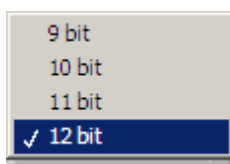
- rozdzielczość czujników temperatury,
- odstęp czasowy między kolejnymi seriami pomiarowymi,
- liczba serii pomiarowych.



Rysunek 16 Zakładka Pomiary programu PKDK

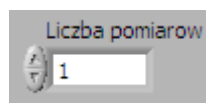
W celu uruchomienia pomiarów należy wcisnąć przycisk *Rozpocznij* (OK).

Wybór rozdzielczości umożliwia rozwijalna lista (dostępna po naciśnięciu na strzałkę):



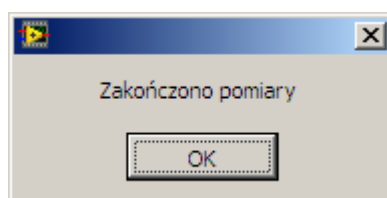
Rysunek 17 Wybór rozdzielczości czujników temperatury

Ustawienie wartości odstępu czasowego i liczby pomiarów możliwe jest poprzez wpisanie wartości z klawiatury. Widoczne po lewej stronie pola kursory, umożliwiają korektę wprowadzonych wartości. Skok wartości w górę lub w dół wynosi 1.



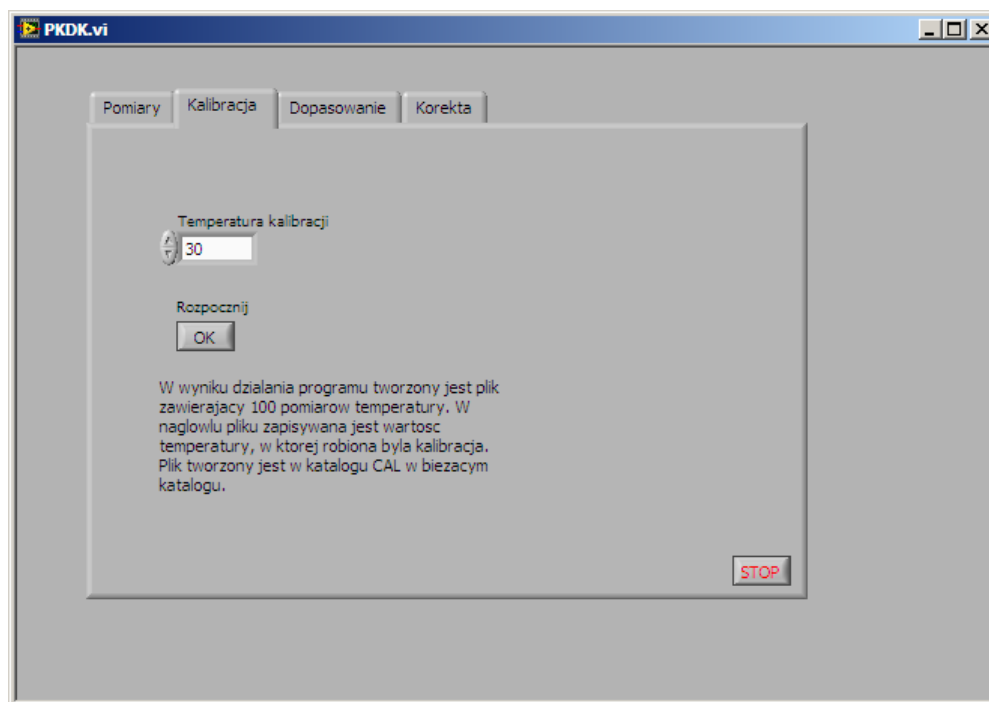
Rysunek 18 Pole wyboru liczby serii pomiarowych

Po zakończeniu pomiarów wyświetlana jest informacja.



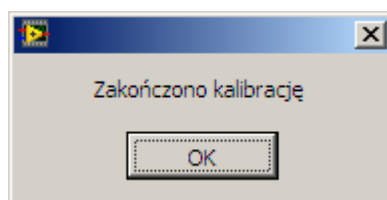
Rysunek 19 Okienko potwierdzające zakończenie pomiarów

4. Na zakładce *Kalibracja* widoczne jest pole liczbowe do wpisywania temperatury, w której wykonywana była kalibracja czujników. Przycisk *OK* służy do uruchamiania kalibracji.



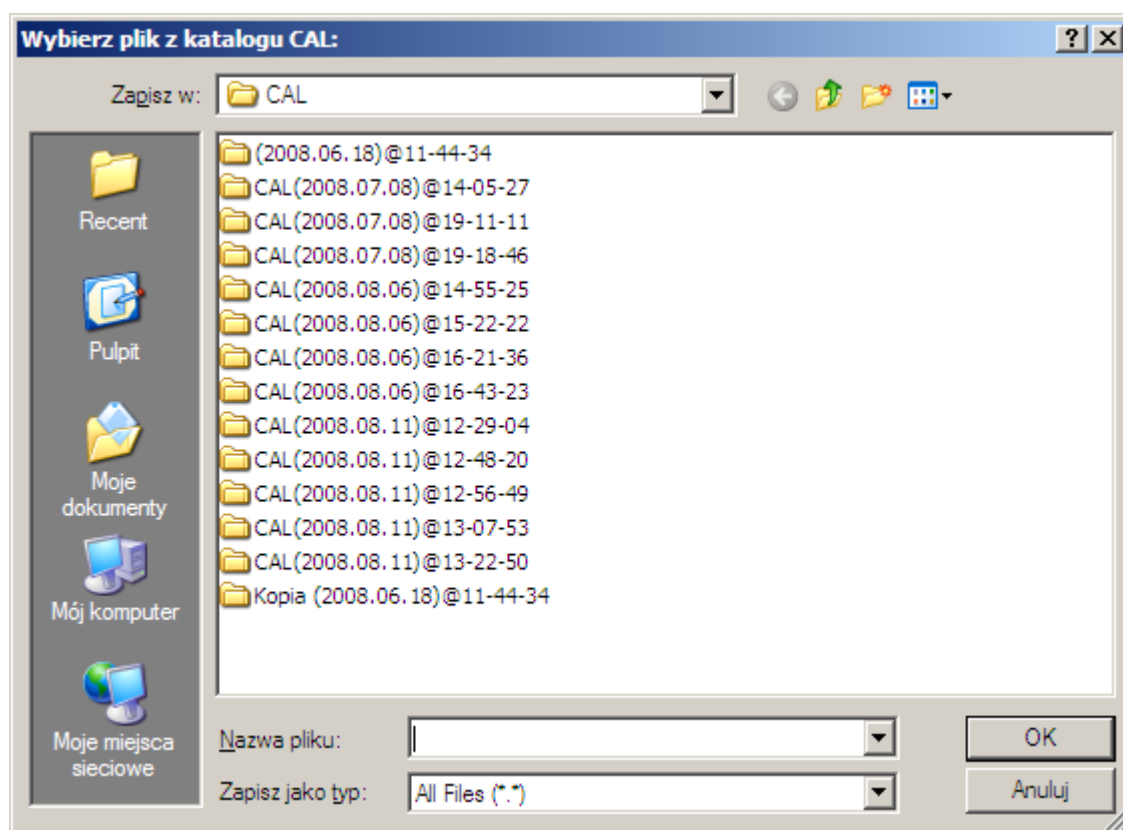
Rysunek 20 Zakładka Kalibracja programu PKDK

Po zakończeniu kalibracji, wyświetlana jest wiadomość:



Rysunek 21 Informacja o zakończeniu kalibracji

5. Zakładka *Dopasowanie* służy do uruchomienia procedury wyznaczania współczynników dopasowania. W pierwszym kroku należy wybrać liczbę wczytywanych plików kalibracyjnych i stopień wielomianu. Rozpoczęcie wyznaczania współczynników rozpoczyna się po wciśnięciu przycisku *OK*. Następnie program pyta użytkownika o wybór plików kalibracyjnych.

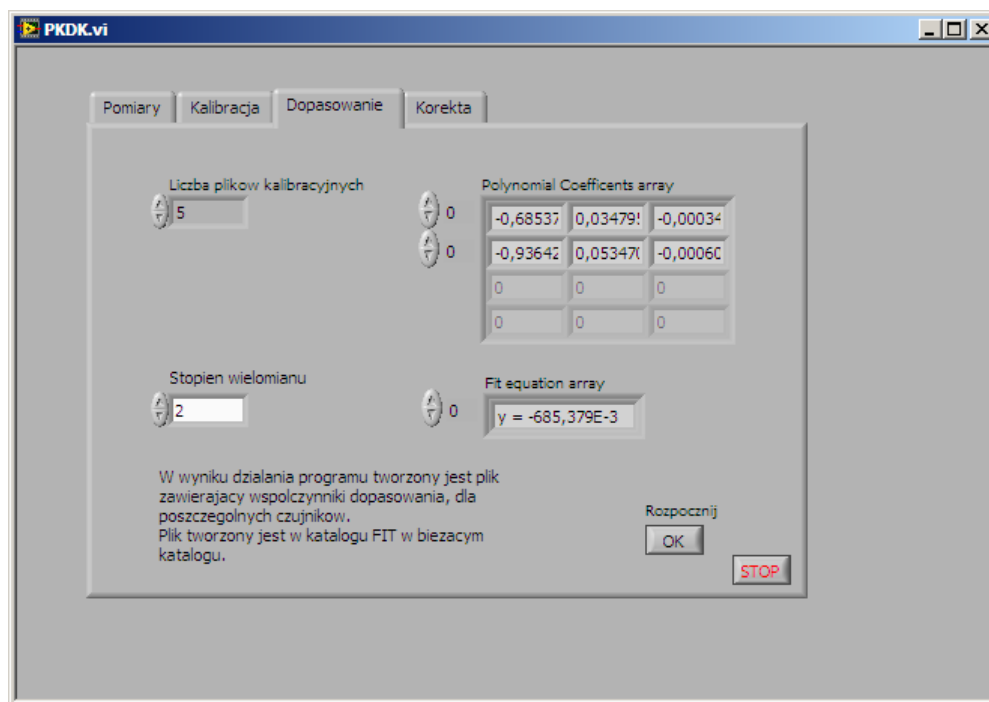


Rysunek 22 Wybór pliku kalibracyjnego

Po zakończeniu działania programu na ekranie widoczne są:

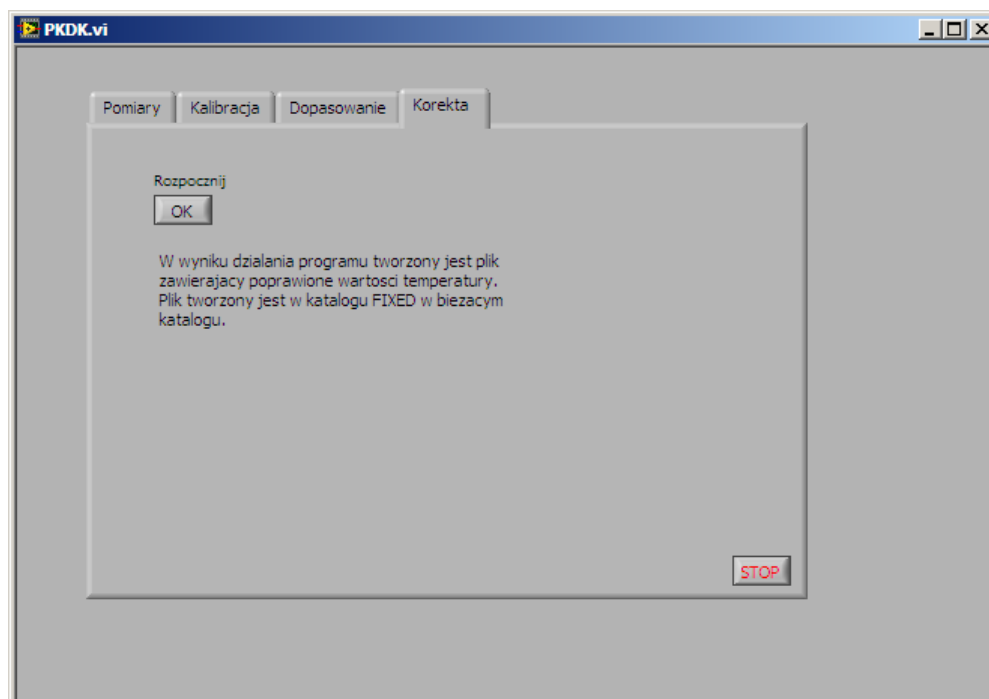
- tablica współczynników dopasowania,
- tablica równań krzywych dopasowania.

W celu przeglądania odpowiednich komórek tablicy, należy skorzystać z kursorów.



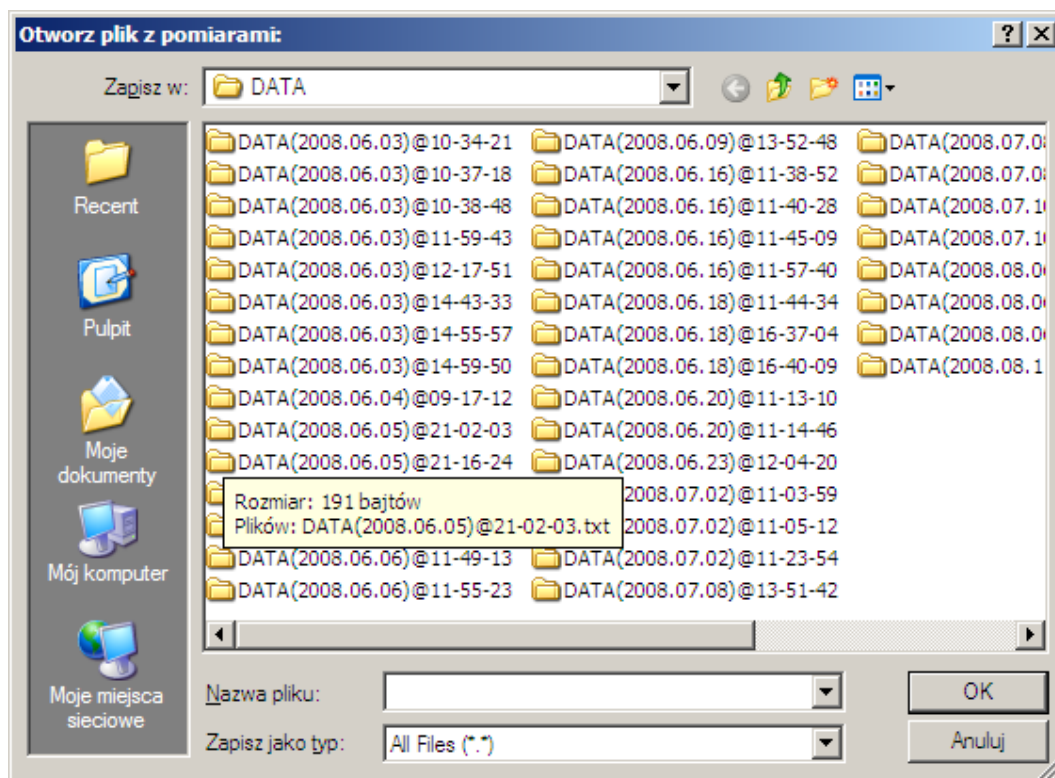
Rysunek 23 Zakładka Dopasowanie programu PDKD

6. Zakładka *Korekta* pozwala poprawić wykonane wcześniej „surowe” pomiary za pomocą współczynników dopasowania. W celu uruchomienia procedury należy wcisnąć przycisk *OK*.

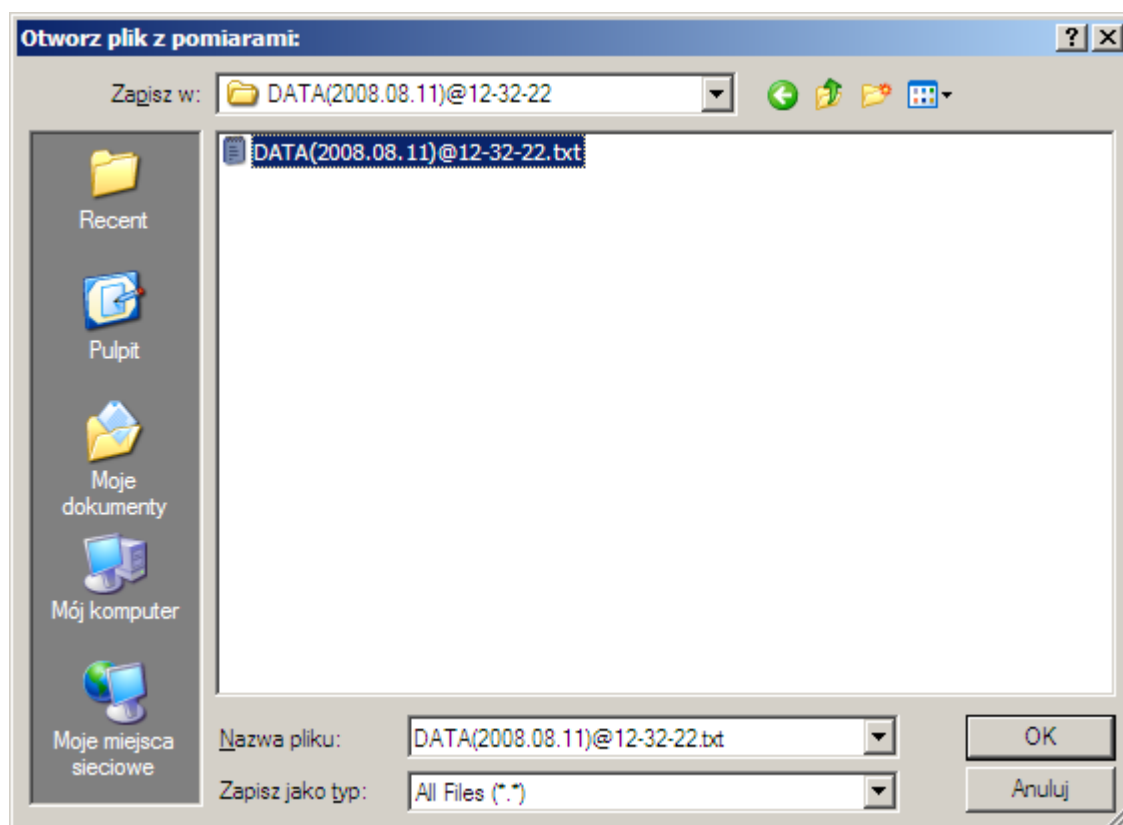


Rysunek 24 Zakładka Korekta programu PDKD

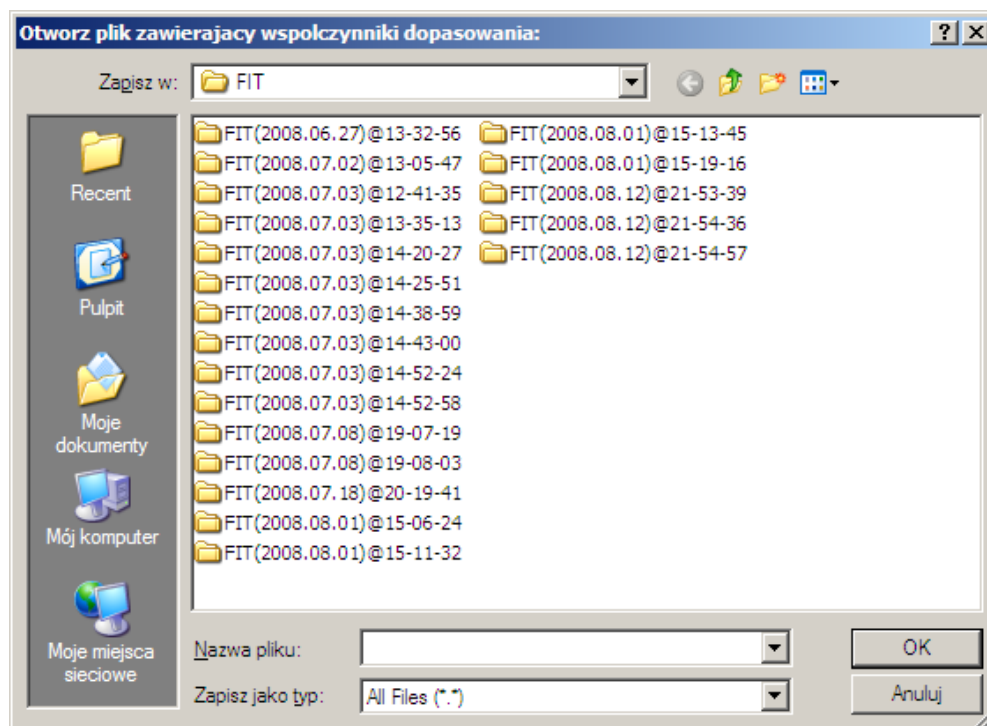
Po uruchomieniu program pyta o wskazanie kolejno pliku z pomiarami a następnie o plik ze współczynnikami dopasowania.



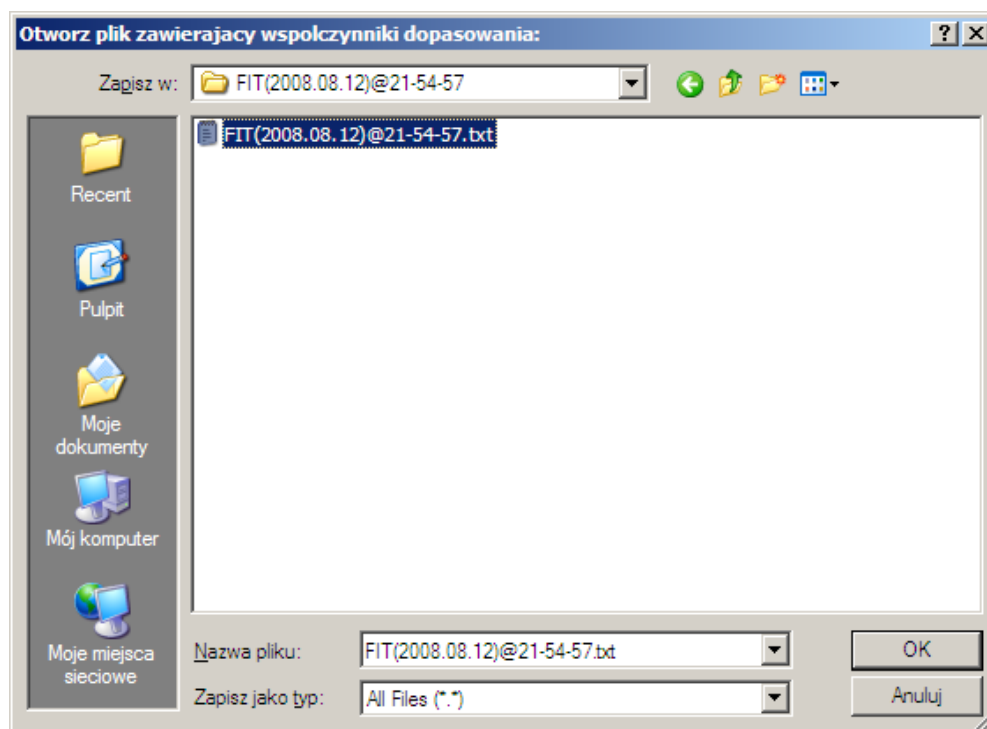
Rysunek 25 Wybór katalogu z plikiem pomiarowym



Rysunek 26 Wybór pliku pomiarowego

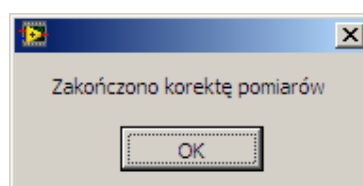


Rysunek 27 Wybór katalogu zawierającego plik ze współczynnikami dopasowania



Rysunek 28 Wybór pliku zawierającego współczynniki dopasowania

Zakończenie korekty pomiarów potwierdzone jest informacją.



Rysunek 29 Okienko zakończenia korekty pomiarów