

Agentowy system roju – możliwości aplikacyjne

Jerzy Tłaga, Waldemar Tłaga

W opracowanym w firmie Alarmtech elektronicznym systemie zabezpieczeń zastosowano architekturę systemu roju wspartą technologią programowania agentowego. Przedstawione podejście pozwoliło zrealizować w pełni dwa podstawowe założenia:

- **budowa systemu możliwa jest z dowolnego zestawu dowolnych komórek w dowolnej topologii,**
- **wszystkie komórki współdzielą energię zasilania roju oraz dostępną informację i zasoby.**

W efekcie uzyskano system o unikalnych właściwościach. Proste w konstrukcji komórki składowe po połączeniu w rój nawiązują automatycznie współpracę, tworzą spójny system roju o możliwościach rosnących z jego skalą. Architektura roju zapewnia prostotę budowy, elastyczność oraz wysoki poziom bezpieczeństwa.

Architektura systemu

W poszukiwaniu optymalnej architektury systemowej dla elektronicznych systemów zabezpieczeń zwrócono uwagę na interesujące wyniki aplikacyjne architektur wywodzących się z badań nad sztuczną inteligencją. W projektowanym systemie założono zastosowanie architektury roju wraz z programowaniem agentowym [1].

Elektroniczny system roju

Podstawowym elementem systemu roju jest komórka. Komórka jest autonomicznym modulem o funkcjonalności określonej przez:

- ilość łącz do innych komórek,
- typ i ilość zasobów wewnętrznych (linie wejścia, wyjścia, itp.),
- obecność lub brak wewnętrznego źródła energii (komórki aktywne i pasywne),
- zespół agentów programistycznych zainstalowany w komórce.

Ilość typów komórek stosowana do budowy rojów nie jest określona. Każda komórka może być połączona z dowolną inną komórką w dowolnej konfiguracji. Komórki łączą się prostymi, zunifikowanymi łączami p2p (*peer-to-peer*), poprzez które współdzielą energię zasilania w roju oraz informację. Nie ma ograniczenia ilości i typów komórek uczestniczących w roju – zależy to wyłącznie od potrzeb aplikacji. System roju może się składać z jednej komórki, kilku, kilkunastu lub dziesiątek różnych komórek połączonych w dowolny, ale optymalny dla danej aplikacji sposób [2], [3].

Programowanie agentowe

Programowanie agentowe to poziom abstrakcji programowania wyższy od programowania obiektowego. Agent jest obiektem obdarzonym autonomią zachowań i do pewnego stopnia inteligencją w działaniu. Każdy agent programistyczny zastosowany w opisywanym systemie posiada następujące cechy:

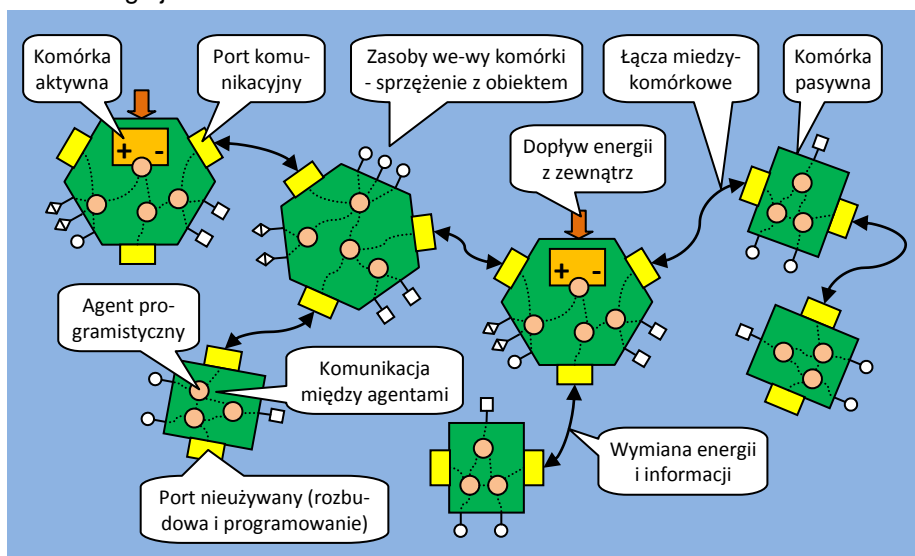
- Autonomiczność - zdolność podejmowania samodzielnych decyzji,
- Komunikatywność - umiejętność komunikacji z innymi agentami,
- Percepcja - zdolność do postrzegania i reagowania na zmiany środowiska i obiektu,
- Mobilność – możliwość przemieszczania się z jednej komórki do drugiej.

Opisywany system jest systemem wieloagentowym (*multi-agent system*), złożonym z wielu komunikujących i współpracujących ze sobą agentów (rys. 1).

Topologie systemowe

Architektura roju zapewnia brak jakichkolwiek ograniczeń w topologii systemowej. Każda komórka może być dołączona do dowolnej innej przy pomocy zunifikowanego 4-przewodowego łącza. Pozwala to na zdecydowaną redukcję kosztów systemu poprzez dopasowanie topologii i zasobów do potrzeb danego obiektu.

Jedynym ograniczeniem jest maksymalna liczebność roju (255 komórek) oraz ilość zasobów tego samego typu w pojedynczej komórce (255).



Rys. 1. Elementy składowe elektronicznego systemu roju z programowaniem agentowym

Nazwa komórki	Ilość portów	Ilość wejść	Ilość wyjść	Zasilacz	Opis komórki
NC388	3	8	8	12V/3A, ViP	Węzeł roju z zasilaczem
NC288PC	2	8	8	12V/3A, ViP	Węzeł roju z zasilaczem + platforma PC
NC284	2	8	4	-	Węzeł roju
NC242	2	4	2	-	Węzeł roju
NC2DD	2	12+1	13+13*	-	Węzeł roju z tablicą synoptyczną LED*
UI242	2	4	2	-	Manipulator wewnętrzny
UE100	1	-	1	-	Manipulator zewnętrzny
TS212	2	1*	2	-	Systemowy czujnik temperatury z rejestratorem
USB100	1	-	-	-	Mostek komunikacyjny USB
GSM142	1	4	2	Akumulator	Węzeł roju z mostkiem komunikacyjnym GSM
LAN142	1	4	2	-	Węzeł roju z mostkiem komunikacyjnym LAN

Tab. 1. Parametry wybranych komórek używanych do budowy roju

Podstawowe cechy systemu

Automatyczna konfiguracja

Każda komórka ma wbudowaną zdolność nawiązywania współpracy z inną komórką. Cecha ta jest aktywowana tylko w momencie tworzenia nowego roju lub wprowadzania zmian w topologii roju istniejącego. Procedura ta jest całkowicie automatyczna i jest inicjowana przy pomocy jednej zworki w jednej z wybranych

komórek.

Po autokonfiguracji zbiór niezależnych połączonych komórek staje się jednym spójnym rojem – zbiorem współpracujących ze sobą komórek o w pełni chronionej integralności.

Współdzielenie energii w roju przy pomocy zasilaczy z funkcją ViP

Komórki aktywne zostały wyposażone w moduły zasilaczy z funkcją ViP umożliwiającą równoległą pracę wielu zasilaczy na jednej linii zasilania.

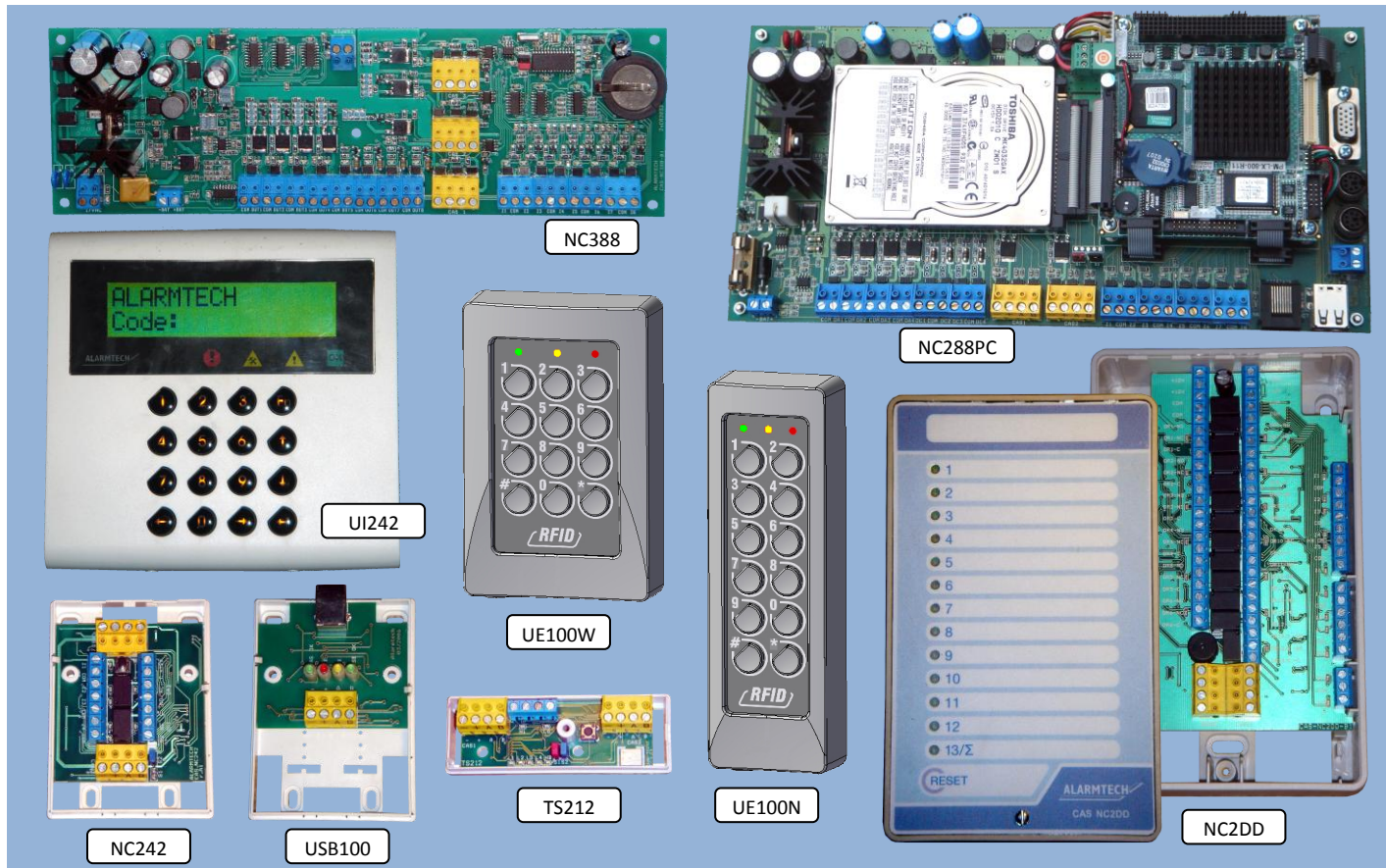
nia. W roju może znajdować się dowolna ilość dowolnie połączonych komórek aktywnych tworząc rozproszone źródło zasilania roju. Komórki aktywne zasilają otaczające je komórki pasywne.

Komórka aktywna przekształca się automatycznie w pasywną w przypadku wyłączenia własnego zasilacza i wraca do stanu aktywnego po ponownym jego włączeniu.

System posiadający komórki aktywne cechuje się bardzo dużą niezawodnością zasilania – komórki aktywne automatycznie optymalizują warunki zasilania. W przypadku awarii lub ataku minimalizują straty systemu próbując wykorzystać wszystkie dostępne źródła energii w systemie do utrzymania jego funkcjonalności na maksymalnym możliwym poziomie [2].

Odporność na ataki

Odporność na ataki jest zagwarantowana przez tajność przesyłania wszystkich danych w systemie. Wszystkie połączenia są szyfrowane unikalnym kluczem. Niezależnie od szyfrowania wszystkie pakiety są



Rys. 2. Przykładowe komórki roju (widok bez obudowy)

uwierzytelniane podpisem cyfrowym.

Struktura roju także podlega ciągłemu monitorowaniu i każda jej zmiana - bez powtórnej autokonfiguracji - jest wykrywana jako naruszenie struktury. Nienaruszalność struktury węzłów komunikacyjnych w roju (współpracujące porty i trakt) jest kontrolowana algorytmem DNA - *Digital Node Authentication*.

Zastosowane rozwiązania zapewniają całkowitą odporność systemu na ataki: bierne, czynne i przez podstawienie [2].

Samonaprawialność

W przypadku ataku uszkodzone komórki są identyfikowane i wykluczane z systemu. Rój broni się w ten sposób przed współpracą z uszkodzonymi komórkami i propagacją uszkodzeń.

Co pewien czas uszkodzone, „chore” komórki otrzymują szansę powrotu do roju pod warunkiem przejścia testów dokumentujących ich „wyzdrowienie”. Przyjęcie „uzdrowionej” komórki z powrotem do systemu roju i podjęcie z nią komunikacji jest poprzedzone testami integralności wszystkich łączy pomiędzy daną komórką a systemem.

Cecha samonaprawialności zwiększa niezawodność systemu i pozwala odzyskiwać zasoby częściowo utracone w wyniku przejściowych ataków lub uszkodzeń.

Elastyczność i skalowalność

Architektura roju zapewnia brak ograniczeń w ilości i rodzaju łączonych komórek oraz dowolność ich połączeń w każdej topologii. Efektem jest prostota rozbudowy i modyfikacji - systemy można budować z dowolnych elementów połączonych w dowolny sposób.

Każda nowa komórka dokłada do istniejącego systemu swoich agentów zwiększając tym samym możliwości systemu. Każdy agent z dowolnej komórki może komunikować się z dowolnym agentem z dowolnej innej komórki, co zapewnia, że zwiększanie skali systemu nie pogarsza parametrów obliczeniowych systemu związanych ze wzrostem jego skali.

Rój jako system wielozadaniowy i wieloprotokółowy

Każdy agent jest obiektem auto-

nomicznym pracującym w systemie czasu rzeczywistego i w trybie wielozadaniowym. Z chwilą połączenia komórek w rój, agenci z poszczególnych komórek zaczynają współpracować ze sobą. Rój staje się w ten sposób wieloprotokółowym systemem wielozadaniowym. Ilość zaangażowanych procesorów i współpracujących agentów rośnie wraz ze skalą systemu zwiększając możliwości przetwarzania danych.

Rój jako platforma wieloaplikacyjna

Poszczególni agenci mają pełną swobodę współpracy i mogą być powiązani w dowolnie złożone zespoły - w tym pracujące niezależnie od siebie. W ten sposób można w jednym systemie roju uruchomić dowolną ilość aplikacji. Mogą to być np. rozdzielne logicznie systemy zabezpieczeń pracujące w jednym fizycznym roju czy też systemy różnego typu (system alarmowy, system kontroli dostępu, automatyka domowa) używające wspólnych zasobów sprzętowych.

Programowanie roju

Programowanie roju możliwe jest na dwóch poziomach:

- Agentowym - zapewniającym dużą elastyczność programowania poprzez indywidualne programowanie agentów,
- Funkcjonalnym - zapewniającym szybkość i prostotę programowania. Dodatkową zaletą tej techniki jest przenaszalność programu

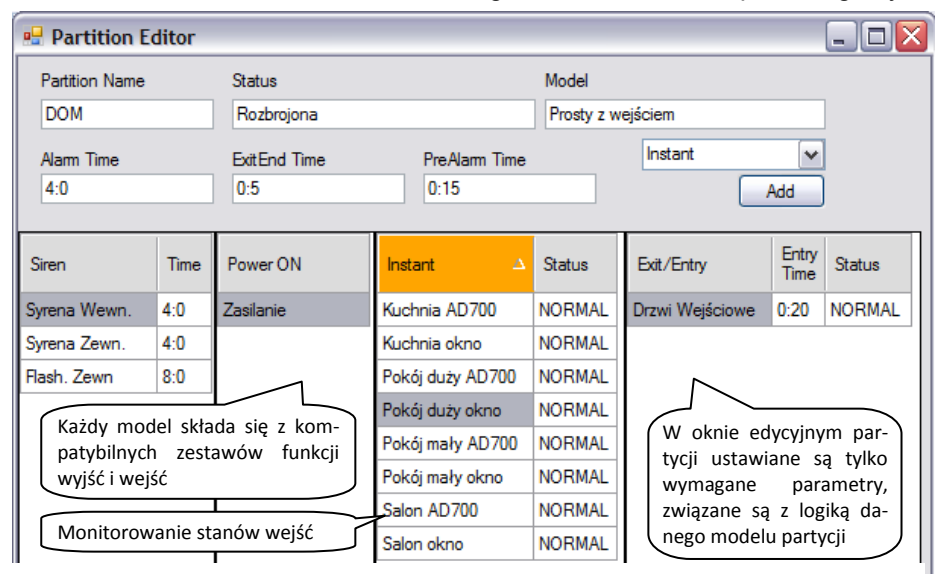
nawet pomiędzy rojami o różnej strukturze.

Do programowania można używać manipulatorów LCD, lokalnych komputerów PC dołączonych przez mostek USB100 lub zdalnych komputerów podłączonych przez sieć LAN.

Programowanie agentowe

Wszystkie nastawy w systemie zmieniane w czasie programowania są przechowywane przez poszczególnych agentów. Programowanie ich jest możliwe przy pomocy manipulatora lub programu komputerowego pracującego w trybie serwisowym, który realizuje dostęp i operacje na dowolnym agencie w systemie. Aby uzyskać dostęp do agenta, najpierw wybiera się moduł, a następnie w menu wybranego modułu należy wybrać agenta. Po wybraniu agenta można wykonać na nim operacje związane z nastawami (np. zmiana typu linii) lub z monitorowaniem stanu jego zasobów (odczyt wartości rezystancji mierzonej przez linię). Może to być np. odczytanie wartości rezystancji linii, lub zmiana typu linii z NC na EOL. Protokół wymiany informacji między manipulatorem a agentami pozwala na współpracę z dowolną ilością i typem agentów obecnych w roju.

Mnogość, różnorodność i rozdystrybuowanie agentów z jednej strony dają dużą elastyczność w programowaniu, z drugiej jednak strony takie rozdysponowanie programowania na poszczególnych



Rys. 3. Programowanie partycji - programator wybiera dane do wyświetlania najważniejsze dla programisty w danym modelu partycji

agentów wydłuża listę parametrów o parametry związane ze strukturą sieci i powiązaniem między agentami szczególnie w dużych systemach.

Programowanie funkcjonalne

Dlatego zastosowano drugi model programowania - funkcjonalny, w którym użytkownik posługuje się tradycyjnymi pojęciami typu partycja, linia, wejście, itd., a struktura i powiązania między agentami zostają ukryte techniką enkapsulacji. Wszystkie parametry związane ze strukturą i rozmieszczeniem agentów zostają automatycznie wygenerowane przez program na podstawie parametrów partycji, linii, itd. oraz na podstawie struktury sieci. Programowanie funkcjonalne można porównać do aplikacji Java: tak jak aplikację Java można uruchomić na dowolnej platformie systemowej i sprzętowej, tak program zapisany w modelu funkcjonalnym można przenosić na dowolne konfiguracje sprzętowe modułów CAS (o ile oczywiście jest wystarczająca ilość zasobów).

Model ten ma dwie podstawowe zalety:

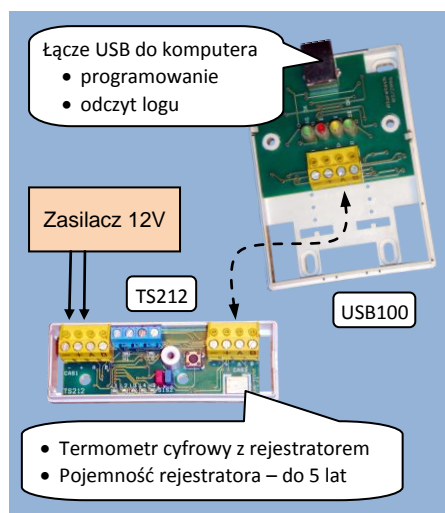
- Użytkownik posługuje się pojęciami prostymi, nieobciążonymi bagażem nastaw związanych ze strukturą sieci lub specyfiką systemu roju,
- Nastawy obiektów takie jak partycja, linia są łatwo przenaszalne z aplikacji na aplikację (np. z dwóch komórek NC242 na jedną NC388). Pozwala to dodatkowo na tworzenie własnych szablonów nastaw, przy pomocy których tworzenie nowych aplikacji może być prawie automatyczne.

Przykłady zastosowań

Każda komórka w systemie roju jest autonomiczna – może pracować samodzielnie. Zapewnia to bardzo dużą skalowalność systemów roju - od poziomu roju składającego się z 1 komórki do kilkuset. Poniższe przykłady pokazują budowę systemów roju o różnej skali i złożoności.

Termometr z rejestratorem (rys 4)

Przykład pokazuje wykorzystanie autonomii komórki. Komórka termometr posiada wszystkie zasoby niezbędne do samodzielnej pracy jako rejestrator temperatury.



Rys. 4. Rejestrator temperatury

Regulator temperatury (rys. 5)

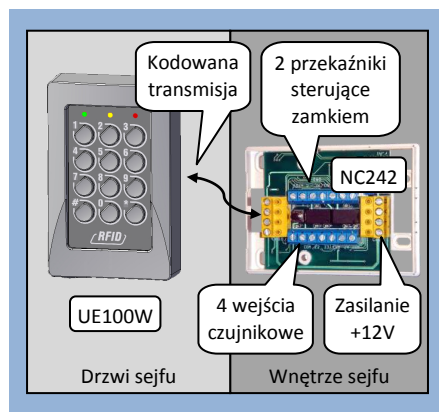
Wykorzystanie 2 wyjść znajdujących się w komórce TS212 do sterowania urządzeniem grzejmym/chłodzącym pozwala przekształcić rejestrator temperatury w regulator temperatury o programowanym zakresie regulacji.

Zamek kodowy (rys. 6), kontrola dostępu (rys. 7)

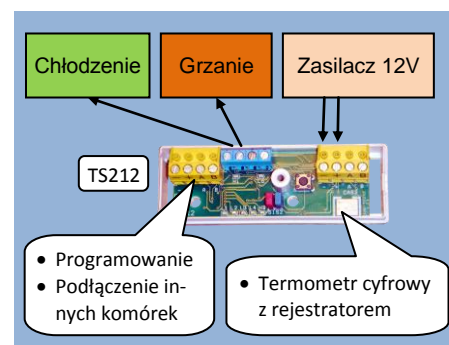
Proste zestawy komórek mogą być wykorzystane do budowy prostych systemów, które jednocześnie dają możliwość rozszerzenia poprzez wolne porty komunikacyjne. Poprzez te same porty te małe aplikacje mogą być dołączane do dużych systemów roju. Na przykładach pokazano 1-2 komórkowe zestawy małych systemów zabezpieczeń z kontrolą dostępu.

Prosty system alarmowy (rys. 8)

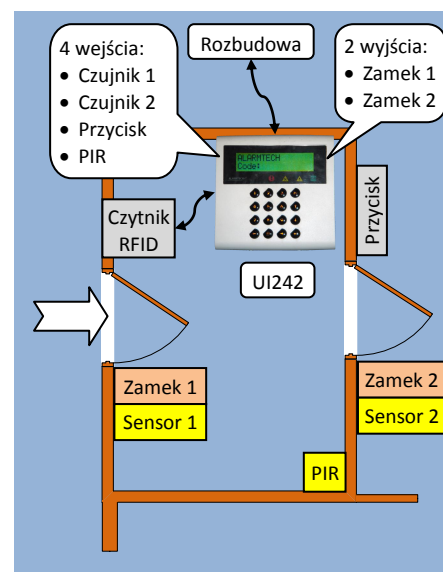
Każda komórka roju oprócz zasobów fizycznych posiada również



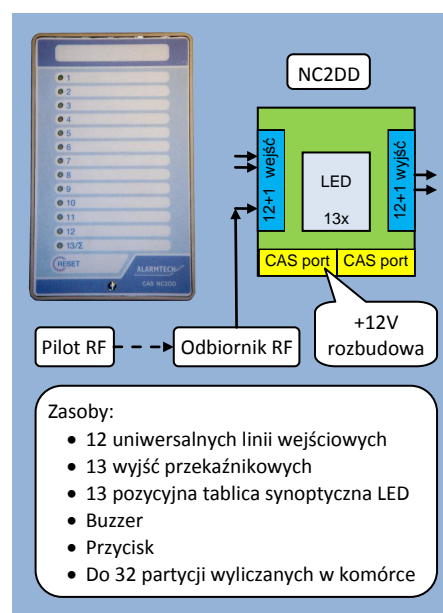
Rys. 6. Zamek kodowy do sejfów



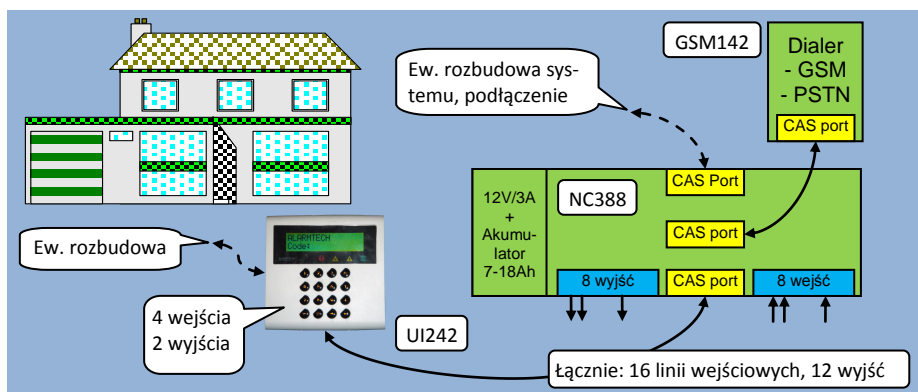
Rys. 5. Regulator temperatury z rejestratorem



Rys. 7. Zabezpieczenie przejścia przez służbę zasoby obliczeniowe niezbędne do wykonywania wszystkich funkcji systemu zabezpieczeń – np. zarządzania



Rys. 8. Prosty system alarmowy



nie i wyliczanie stanu partycji. Pozwala to na tworzenie systemów zabezpieczeń o dowolnej skali i złożoności. Pojedyncza komórka jest w stanie zarządzać 32 partycjami - dołożenie do roju kolejnych komórek odpowiednio zwiększa ilość dostępnych zasobów, w tym partycji. Przykład pokazuje budowę systemu alarmowego na bazie 1 komórki.

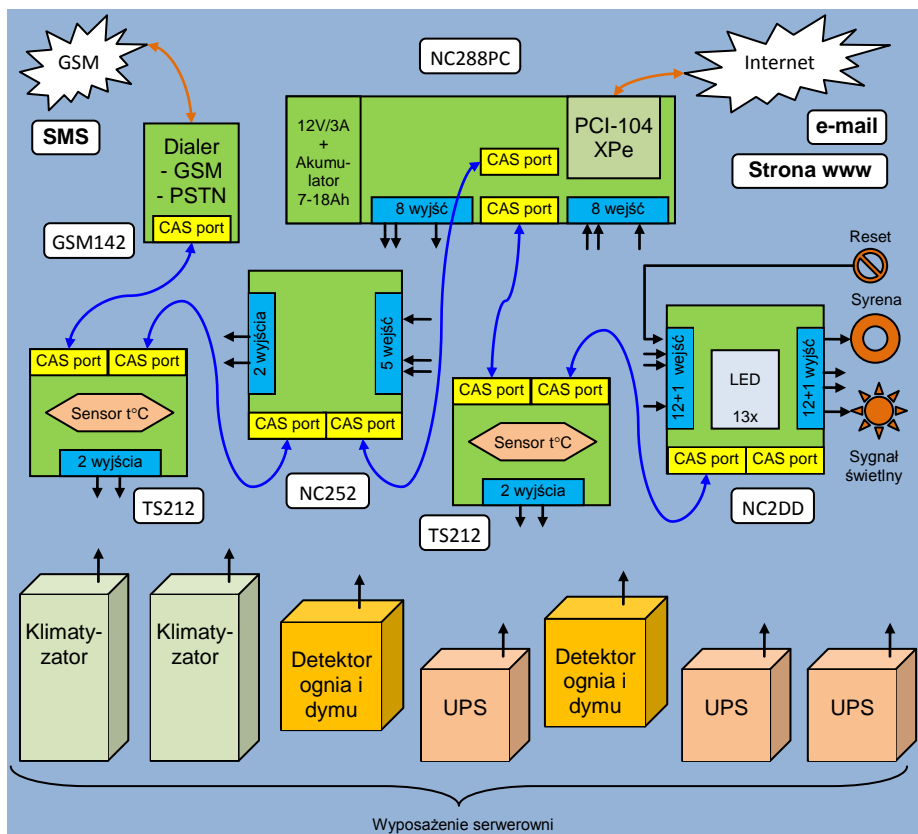
System alarmowy z powiadamianiem
(rys. 9)

Podstawową komórką używaną do budowy systemów zabezpieczeń domów, mieszkań, biur jest najczęściej NC388. Posiada ona wbudowany zasilacz 12V o wydajności łącznej 3A i może współpracować z

akumulatorami o pojemności od 7Ah do 18Ah. Zasilacz komórki spełnia wymagania normy EN 50131-6 w klasie 3 i 4 (klasa 4 wymaga odpowiedniej obudowy). Zapewnia to budowę systemów o wysokiej klasie bezpieczeństwa. Rój 3 komórek (NC388 + manipulator LCD UI242 + dialer GSM GSM142) tworzy system o zasobach wystarczających do ochrony domu, mieszkania lub biura na wysokim poziomie.

System nadzoru obiektu monitoro-
wany przez Internet (rys. 10, rys. 11)

Możliwość nadzoru stanu obiektu przez Internet jest szczególnie cenna dla firm posiadających swoje obiekty rozrzucone na dużym obszarze.



Rys. 10. System nadzoru stanu obiektu z powiadamianiem i monitorowaniem przez Internet

Przykładem mogą być systemy utrzymania parametrów klimatycznych w serwerowniach. Wymagają one regularnej obsługi (wymiana czynników) oraz ciągłego nadzoru parametrów. Takie właśnie zastosowanie systemów roju wymagające komunikacji na platformie IP zostało wybrane jako aplikacja testowa.

Na bazie komórki NC388 skonstruowano komórkę NC288PC z wbudowanym komputerem klasy „PC embedded” wykonanym w standardzie PCI-104. Komputer ten pracuje pod kontrolą systemu operacyjnego „MS Windows XP’embedded” tworząc otwartą platformę aplikacyjną. Ułatwia to zdecydowanie integrację systemu roju z otoczeniem programistycznym.

Z punktu widzenia roju komputer jest kolejną komórką ze swoim zestawem agentów. Jednym z nich jest agent automatycznej kreacji strony sieciowej wizualizującej rój oraz monitorującej jego parametry. Rozpoznaje on strukturę roju, identyfikuje komórki i ich zasoby oraz tworzy ich graficzną reprezentację w formie strony www. Wybrani użytkownicy są informowani o stanie obiektu przy pomocy protokołów HTTP (strona sieciowa), SNMP i SMTP (e-mail).

Systemy szkieletowe (rys. 12)

Każda komórka roju jest autonomiczna w zakresie posiadanych zasobów. Każda komórka posiada także silnik logiczny pozwalający zarządzać stanami do 32 partycji. Stanowi zatem kompletny mikrosystem alarmowy mogący pracować samodzielnie; może też być łączona w większe systemy.

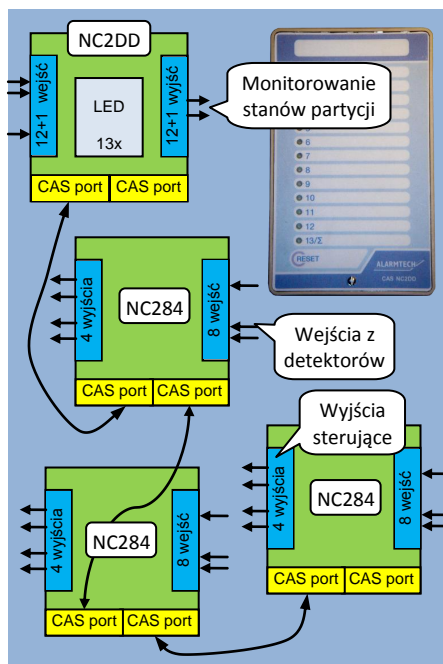
Taka koncepcja komórki roju pozwala na budowę systemów alarmowych w postaci szkieletowej z użyciem prostych i tanich komórek. Wejścia komórek używane są do zbierania stanu detektorów, część wyjść do sterowania odpowiednimi zasobami (np. sygnalizatory), a część wyjść może być użyta do monitorowania stanów partycji.

Modernizacja systemów alarmowych bez zmian tras kablowych

Ze względu na dużą elastyczność system roju może też być przydatny w przypadku konieczności modernizacji systemów. Ponieważ rój może pracować w dowolnej topologii, pozwala to na modernizację starych systemów bez konieczności kosztownej przebudowy tras kablowych czy zmian w parametrach linii detektorów.

W przypadku modernizacji systemów o scentralizowanej konstrukcji, wystarczy starą centralkę alarmową wymienić na rój zapewniający taką samą ilość i typ zasobów (wejścia, wyjścia).

Dla systemów preferujących pra-



Rys. 12. Szkielet systemu alarmowego
- 36 wejść, 12 wyjść, 6 partycji

cę z modułami rozszerzeń wystarczy wymienić moduły na równoważne im pod względem wymaganych zasobów komórki lub zestawy komórek.

Wymiana tak jest zawsze możliwa, ponieważ zbiór topologii roju zawiera w sobie wszystkie topologie systemów interfejsowych.

Systemy alarmowe z detektorami o różnych parametrach linii (rys. 13)

Wszystkie linie wejściowe we wszystkich komórkach pracują w trybie analogowym z programowalnym stanem rezystora polaryzującego (ON/OFF). Agent linii mierzy napię-

cie na wejściu i prąd płynący przez rezystor polaryzujący oraz wylicza aktualną wartość rezystancji widzianą przez wejście linii. Dzięki takiej konfiguracji wejścia możliwa jest praca w dowolnej konfiguracji linii, w tym w konfiguracji niestandardowej zdefiniowanej przez użytkownika. Ma to szczególne znaczenie przy współpracy np. z różnego typu czujkami ognia i dymu.

Oznacza to także, że każda linia dowolnej komórki roju może współpracować z detektorem skonfigurowanym na inną wartość rezystora EOL.

Modernizacja systemów bez zmian parametryzacji detektorów

Omówiona wyżej cecha pozwala wykorzystywać komórki roju do modernizacji systemów bez potrzeby zmiany konfiguracji lub parametrów detektorów. Wystarczy zaprogramować odpowiednio wejścia komórek dopasowując je do zastanych parametrów pętli lub detektorów.

Podsumowanie

Przedstawione przykłady nie wyczerpują wszystkich możliwych zastosowań systemów roju. W podanych przykładach skoncentrowano się w większości na zastosowaniach w systemach alarmowych.

System roju z programowaniem agentowym można łatwo stosować w innych zbliżonych dziedzinach, takich jak automatyka domowa, systemy kontroli dostępu czy systemy zintegrowane.

Literatura

- [1] Jerzy Tłaga, Waldemar Tłaga, "Self-organizing electronic hive system for security applications", IV International Congress of Technical Diagnostics, zgłoszenie, 2008
- [2] Jerzy Tłaga, Waldemar Tłaga, "Sammoorganizujący się system roju – nowe podejście w projektowaniu systemów zabezpieczeń (cz. 1), Ochrona Mienia i Informacji, nr 5/2007, str. 12-17
- [3] Jerzy Tłaga, Waldemar Tłaga, "Sammoorganizujący się system roju – nowe podejście w projektowaniu systemów zabezpieczeń (cz. 2), Ochrona Mienia i Informacji, nr 6/2007, str. 38-42

Hardware Name	User Name	Type	Sensitivity [s]	Prior	Inhib	3	System Line	Status
3-Line 1	Drzwi Wejściowe	EOL 1.1 kOhm	0.10	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
3-Line 2	Pokój duży okno	2EOLNC 2*(1.1) kOhm	0.20	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
3-Line 3	Pokój duży AD700	EOL 3.3 kOhm	0.40	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
3-Line 4	Kuchnia okno	2EOLNC 2*(3.3) kOhm	1.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
3-Line 5	Kuchnia AD700	EOL 5.6 kOhm	1.30	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
3-Line 6	Salon okno	2EOLNC 2*(5.6) kOhm	2.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
3-Line 7	Salon AD700	EOL 4.7 kOhm	3.00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
3-Line 8	Pokój mały okno	2EOLNC 2*(4.7) kOhm	4.00	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
2-Line 1	Pokój mały AD700	Custom	20	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
2-Line 2	2-Line 2	NO	0.40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
2-Line 3	2-Line 3	NO	0.40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL
2-Line 4	2-Line 4	NO	0.40	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NONE	NORMAL

Rys. 13. Przykład programowania różnych parametrów pracy linii