CNBOP-PIB

WYMAGANIA CNBOP-PIB

Certyfikacja oprogramowania – ocena oprogramowania urządzeń przeciwpożarowych

(wydanie: pierwsze, 07 sierpień 2024 r.)

CENTRUM NAUKOWO-BADAWCZE
OCHRONY PRZECIWPOŻAROWEJ
im. Józefa Tuliszkowskiego
PAŃSTWOWY INSTYTUT BADAWCZY
ul. Nadwiślańska 213, 05-420 Józefów

TWÓJ PARTNER W
BEZPIECZEŃSTWIE

Dokument opracował zespół autorski w składzie:

mgr inż. Michał Pietrzak
mgr inż. Wojciech Gągała

Projekt okładki: Marta Iwańska
Projekt graficzny zawartości: Marta Iwańska
Grafiki na okładce: cleanpng.com

Wydawca:

Centrum Naukowo-Badawcze Ochrony Przeciwpożarowej
im. Józefa Tuliszkowskiego
Państwowy Instytut Badawczy
05-420 Józefów, ul. Nadwiślańska 213
tel. (22) 76 93 300
www.cnbop.pl
e-mail: cnbop@cnbop.pl

Wydanie I, sierpień 2024, Józefów

Spis treści

1. Informacje ogólne

Urządzenia i instalacje przeciwpożarowe są jednymi z podstawowych środków zapewniających bezpieczeństwo przeciwpożarowe obiektów budowlanych. Poprawnie dobrane i zainstalowane systemy pozwalają na niezwłoczne wykrycie zagrożenia pożarowego oraz umożliwiają sprawną, zorganizowaną i bezpieczną ewakuację użytkowników z zagrożonych obszarów, zapewniają także możliwość podjęcia gaszenia we wczesnej fazie rozwoju pożaru. Dodatkowo, w celu ułatwienia nadzoru nad wykorzystanymi systemami i urządzeniami można zastosować w obiekcie system integrujący urządzenia przeciwpożarowe, który wspiera procesy decyzyjne w obszarze ochronny przeciwpożarowej obiektu budowlanego. Mając do dyspozycji dużą ilość wyrobów wykorzystywanych do osiągnięcia wymaganego poziomu bezpieczeństwa pożarowego niezwykle istotną kwestią w celu poprawnego działania wszystkich urządzeń jest odpowiednie oprogramowanie. Zastosowane oprogramowanie bezpośrednio odpowiada za poprawną pracę wszystkich urządzeń zaczynając od central sygnalizacji pożarowej, central sterujących urządzeniami przeciwpożarowymi, systemów integrujących czy sterowanych programowo elementów detekcyjnych. Mając na uwadze znaczącą rolę oprogramowania w zachowaniu bezpieczeństwa obiektów budowlanych i ich użytkowników, przydatnym może okazać się zweryfikowanie zastosowanego oprogramowania w celu potwierdzenia, że spełnia ono minimalny poziom wymagań ustanowiony w normach zharmonizowanych lub innych dokumentach stanowiących kryteria oceny w procesach certyfikacyjnych. W niniejszych wymaganiach przedstawiono czynności, kryteria i obszary oceny w zakresie potwierdzenia jakości przedmiotowego oprogramowania w procesie certyfikacji dobrowolnej na potrzeby wydania „certyfikatu oprogramowania urządzenia przeciwpożarowego”.

2. Definicje

Centrala dźwiękowego systemu ostrzegawczego – część składowa dźwiękowego systemu ostrzegawczego, która generuje i nadaje sygnały alarmu głosowego do linii głośnikowych w sytuacji, gdy otrzymuje sygnały alarmowe z systemu sygnalizacji pożarowej i/lub z elementów ręcznej obsługi.

Centrala sygnalizacji pożarowej – podzespół systemu sygnalizacji pożarowej, przez który inne podzespoły mogą być zasilane energią, i który jest stosowany między innymi do:

- odbierania sygnałów z przyłączonych czujek i/lub ręcznych ostrzegaczy pożarowych,
- określania, który z tych sygnałów oznacza stan alarmu pożarowego,
- sygnalizowania akustycznie i optycznie każdego takiego stanu alarmu pożarowego,
- wskazywania miejsca zagrożenia.

Dane chwilowe – zmienne dane podlegające czasowej, automatycznej lub ręcznej modyfikacji podczas pracy.

Dane obiektowe – zmienne dane, niezbędne do pracy urządzenia przeciwpożarowego w określonej konfiguracji instalacji.

Dynamiczne zarządzanie pamięcią – metoda umożliwiająca programowi przydzielanie i zwalnianie pamięci w zależności od zmieniających się potrzeb programu.

Komendy sterujące – komendy umożliwiające zmianę stanu aktualnie wykonywanego kodu programu.

Oprogramowanie – całość instrukcji i procedur (programów) oraz powiązanych z nimi danych umożliwiających urządzeniom wykonywanie określonych funkcji.

Pamięć nieulotna – elementy pamięci, które nie wymagają obecności źródła energii w celu zachowania ich zawartości.

Pola użytkowe – element urządzenia, mechaniczny (np. suwak) lub elektroniczny (np. fragment okna), umożliwiający sterowanie pracą oprogramowania.

Poziom dostępu – definiowany przez producenta zakres uprawnień grupy użytkowników do korzystania z określonych funkcji oprogramowania.

Stan bezpieczeństwa – definiowany przez producenta stan, w którym w przypadku uszkodzenia realizacji programu urządzenie nie powoduje nieprawidłowego działania obowiązkowych wyjść, ani nie daje użytkownikowi fałszywego odczucia, że urządzenie pozostaje w stanie roboczym.

System integrujący urządzenia przeciwpożarowe – narzędzie wspomagające kontrolę i obsługę zastosowanych w obiekcie budowlanym systemów bezpieczeństwa pożarowego.

3. Zakres badań

Niniejsze wymagania obejmują badania oprogramowania dla:

1. Central sygnalizacji pożarowej (CSP),
2. Central dźwiękowego systemu ostrzegawczego (CDSO),
3. Central sterujących (stosowanych do oddymiania, gaszenia, sterowania oprawami),
4. Systemów integrujących urządzenia przeciwpożarowe (SIUP),
5. Innych urządzeń przeciwpożarowych.

Możliwość certyfikacji oprogramowania powyższych urządzeń będzie podlegać indywidualnemu rozpatrzeniu.

Dla każdego procesu, badania oprogramowania będą prowadzone w oparciu o indywidualnie przygotowany zbiór wymagań utworzony przy uwzględnieniu mających zastosowanie elementów przedstawionych poniżej, w punktach od piątego do dwunastego. Powyższe wymagania będą opracowywane na podstawie dokumentacji oprogramowania dostarczonej przez Producenta, stanowiącej załącznik do wniosku o udzielenie certyfikacji.

Badania oprogramowania będą prowadzone na dostarczonej do CNBOP-PIB próbce badawczej przygotowanej i skonfigurowanej przez Producenta w uzgodnieniu z CNBOP-PIB. Konfiguracja testowa powinna umożliwiać przeprowadzenie badań dla wszystkich wymagań zidentyfikowanych dla wyrobu.

4. Kategorie oceny

Badania i ocena prowadzona jest w oparciu o zadeklarowane przez Producenta kategorie wskazane w punkcie 6 oraz obligatoryjne badania opisane w punktach od 7 do 12. Do każdej z kategorii zgłoszonego oprogramowania tworzone są dedykowane wymagania w formie „przypadków testowych" uwzględniając dokumentację oraz funkcje zgłoszonego oprogramowania. Potwierdzeniem spełniania niniejszego dokumentu w tym wszystkich określonych badań w zakresie powyższych kategorii, będzie wydany przez CNBOP-PIB „***Certyfikat oprogramowania dla urządzenia przeciwpożarowego***", ważny przez okres 5 lat. Zapisy certyfikatu określają podstawowe dane dotyczące wnioskodawcy, oprogramowania oraz identyfikują funkcje oprogramowania, które zostały potwierdzone w badaniach wraz z opisem sposobu ich realizowania.

Na etapie procesu certyfikacji oprogramowania Jednostka Certyfikująca może uwzględniać wyniki badań przeprowadzone w laboratoriach CNBOP-PIB. Możliwość uznania powyższych wyników będzie podlegać indywidualnej analizie.

5. Diagnostyka części sprzętowej

Przed rozpoczęciem badań funkcjonalnych należy dokonać diagnostyki części sprzętowej w celu weryfikacji poprawnego działania części sprzętowej urządzenia, na którym zainstalowane jest badane oprogramowanie. Weryfikacja powinna zostać dostosowana do rodzaju sprzętu, na którym oprogramowanie ma być instalowane i powinna obejmować, w zależności od konfiguracji, takie elementy jak:

a) weryfikacja poprawnego uruchomieniu sprzętu po dostarczeniu zasilania,
b) weryfikacje działania wszystkich przycisków/elementów sterujących (strzałki, klawiatura, potencjometry, przyciski funkcyjne np. anuluj, resetuj, akceptuj, itp.),
c) weryfikacje działania pól wyświetlacza (poprawna praca wyświetlacza, brak uciętych pól obsługi/informacji),
d) weryfikacje działania diod LED,
e) weryfikacja działania sygnałów dźwiękowych,
f) weryfikacja działania obligatoryjnych urządzeń pomocniczych (klawiatura komputerowa, mysz, mikrofon).

6. Badania funkcjonalne oprogramowania

Badania funkcjonalne oprogramowania mają na celu weryfikację poprawnej realizacji zainicjowanych i predefiniowanych czynności poprzez oprogramowanie. Weryfikacja powinna zostać dostosowana do rodzaju oprogramowania i w zależności od konfiguracji obejmuje:

a) poprawność logowania / dostępu użytkowników,
b) zarządzanie użytkownikami (tworzenie usuwanie użytkowników, resetowanie haseł logowania, zmiany poziomów dostępu),
c) poprawność działania pól użytkowych (przycisków, suwaków, itp.),
d) opcje sortowania i/lub wyszukiwania,

e) możliwości dodawania, edytowania i usuwania danych,

f) postępowanie z błędnie wpisanymi danymi (wartości, format, zakres),

g) opcje automatycznego wylogowania (np. po określonym czasie bezczynności),

h) interpretowanie czynności z różnych poziomów dostępu,

i) sprawdzenie wybranych funkcji oprogramowania opisanych w dokumentacji producenta,

j) nawigacja po oprogramowaniu (czy odnośniki kierują do dobrych zakładek/miejsc),

k) wersje językowe (ustawienie języka, weryfikacja błędów językowych i komunikatów w Polskiej wersji językowej),

l) poprawna identyfikacji logów, historii zdarzeń,

m) identyfikacja zainstalowanej wersji oprogramowania,

n) weryfikacja poprawności importowania danych zewnętrznych (danych obiektowych),

o) weryfikacja poprawnego generowania danych na panelach typu multi-screen,

p) zachowanie danych po zaniku zasilania (czy wgrane dane są utrzymywane w pamięci i nie ulegają uszkodzeniu lub skasowaniu).

7. Dokumentacja oprogramowania

7.1. Dokumentacja przygotowana przez Producenta powinna umożliwiać zapoznanie się z budową oprogramowania i powinna być przedłożona do weryfikacji. Dokumentacja ta powinna zawierać co najmniej następujące informacje:

a) opis funkcjonalny realizacji głównego programu, uwzględniający:

- ❖ zwięzły opis każdego modułu i wykonywanego przez niego zadania,
- ❖ opis współpracy modułów,
- ❖ opis sposobu wywoływania modułów, łącznie z obsługą przerwań,
- ❖ ogólną hierarchię programu,

Funkcjonalny opis przebiegu głównego programu powinien być objaśniony z użyciem jasnej metodologii, odpowiedniej do charakteru oprogramowania, np. graficznego przedstawienia budowy systemu, przepływu danych i przepływu komend sterujących.

b) opis, które obszary pamięci są używane do przechowywania programu, danych obiektowych i danych chwilowych,

Gdy stosowane jest dynamiczne zarządzanie pamięcią, należy wprowadzić separację pomiędzy programem, danymi obiektowymi i danymi chwilowymi oraz należy to opisać w połączeniu z metodą przydziału pamięci.

c) opis współpracy oprogramowania ze sprzętem, na którym ma zostać zainstalowane.

7.2. Szczegółowa dokumentacja konstrukcyjna powinna zawierać co najmniej następujące elementy:

a) opis każdego modułu programu, tak jak jest to wprowadzone do kodu źródłowego programu, zawierający następujące informacje dotyczące:

- ❖ nazwy modułu;
- ❖ informacji dotyczących daty i/lub wersji;
- ❖ opisu wykonywanych zadań;
- ❖ opisu interfejsów obejmującego rodzaj przekazywanych danych, zakres ważności danych i sprawdzanie ważności danych;

b) wykaz kodów źródłowych, włącznie ze wszystkimi ogólnymi i lokalnymi zmiennymi, zastosowane stałe i etykiety, oraz wystarczający komentarz umożliwiający poznanie przebiegu programu,

c) szczegóły wszystkich narzędzi programowych, wykorzystywanych do przygotowania programu (np. narzędzia projektowe wysokiego poziomu, kompilatory, assemblery itp.).

8. Budowa oprogramowania

Budowa oprogramowania powinna spełniać następujące wymagania:

a) oprogramowanie powinno mieć strukturę modułową,

b) budowa interfejsów dla danych generowanych ręcznie i automatycznie nie powinna pozwalać na pojawianie się błędów w realizacji programu,

c) w programie powinny być zastosowane sposoby zapobiegające blokowaniu się systemu.

9. Nadzorowanie programu

9.1. Realizacja programu powinna być nadzorowana. Jeżeli procedury związane z głównymi funkcjami programu przestaną być realizowane, wówczas należy spełnić jedno lub oba następujące wymagania:

a) oprogramowanie powinno sygnalizować uszkodzenie systemu,

b) urządzenie powinno wejść w stan uszkodzenia i sygnalizować uszkodzenia odpowiednich nadzorowanych funkcji.

9.2. Jeżeli program jest realizowany przez jeden procesor, wówczas wykonywanie procedur należy nadzorować za pomocą urządzenia nadzorującego.

9.3. Jeżeli program jest realizowany przez więcej niż jeden procesor, wówczas wykonanie procedur należy nadzorować w każdym procesorze. Urządzenie nadzorujące powinno być związane z jednym lub wieloma procesorami i co najmniej jeden taki procesor powinien nadzorować działanie dowolnego procesora, nie związanego z takim urządzeniem nadzorującym.

9.4. Urządzenie nadzorujące powinno mieć podstawę czasu niezależną od podstawy nadzorowanego systemu. Działanie urządzenia nadzorującego oraz sygnalizowanie uszkodzenia nie powinny być uniemożliwione przez błąd w realizacji programu nadzorowanego systemu.

9.5. Elementy, które uległy uszkodzeniu, powinny wejść w stan bezpieczeństwa nie później, niż nastąpi zasygnalizowanie uszkodzenia. Ten stan bezpieczeństwa nie powinien powodować fałszywego uruchomienia niezamierzonych działań urządzenia.

10. Przechowywanie programów i danych

10.1. Kod programu powinien być utrzymywany w pamięci, która powinna być zdolna do ciągłej, nieodświeżanej i niezawodnej pracy w okresie co najmniej 10 lat. Dodatkowo, w stosunku do programu powinny mieć zastosowanie następujące wymagania:

a) program powinien być utrzymywany w nieulotnej pamięci i może być wpisywany przez osobę o odpowiednim poziomie dostępu,

b) powinna istnieć możliwość identyfikacji wersji programu.

10.2. W stosunku do danych obiektowych powinny być spełnione następujące wymagania:

a) zmiana specyficznych danych obiektowych powinna być możliwa tylko przez osobę o odpowiednim poziomie dostępu,

b) zmiana danych obiektowych nie powinna mieć wpływu na strukturę programu,

c) jeżeli w pamięci ulotnej są przechowywane dane obiektowe, powinny być one zabezpieczone przed utratą podczas zaniku napięcia zasilania przez zastosowanie rezerwowego źródła energii, które może być odłączone od pamięci przez osobę o odpowiednim poziomie dostępu i które jest zdolne do utrzymania zawartości pamięci co najmniej przez 2 tygodnie,

d) jeżeli takie dane są przechowywane w pamięci umożliwiającej zapis i odczyt, wówczas powinien istnieć mechanizm, który będzie zapobiegał wpisowi do pamięci danych podczas normalnej pracy przez osobę bez odpowiedniego poziomu dostępu, tak aby jej zawartość mogła być chroniona podczas uszkodzenia realizacji programu,

e) powinna istnieć możliwość albo odczytywania, albo przeglądu specyficznych danych obiektowych przez osobę o odpowiednim poziomie dostępu, lub specyficzne dane obiektowe powinny mieć nadaną wersję odniesienia, która powinna być aktualizowana po wprowadzeniu każdej zmiany,

f) jeżeli specyficzne dane obiektowe mają wersję odniesienia, to powinna istnieć możliwość ich identyfikacji przez osobę o odpowiednim poziomie dostępu.

11. Nadzorowanie zawartości pamięci

11.1. Zawartość pamięci zawierających specyficzne dane obiektowe powinna być automatycznie sprawdzana w odstępach nie przekraczających 1 h. Urządzenie sprawdzające powinno sygnalizować uszkodzenie systemu, jeżeli zostanie wykryte zniekształcenie zawartości pamięci.

12. Działanie w przypadku uszkodzenia systemowego

12.1. Oprogramowanie powinno wprowadzić stan uszkodzenia gdy zostaną odebrane sygnały, które po niezbędnym przetworzeniu są interpretowane jako uszkodzenia.

12.2. Oprogramowanie powinno być zdolne do jednoczesnego rozpoznawania wszystkich występujących uszkodzeń.

12.3. Oprogramowanie powinno wprowadzić stan uszkodzenia w ciągu 100 sekund od zaistnienia uszkodzenia lub odebrania sygnału uszkodzeniowego.

12.4. Uszkodzenia powinny być sygnalizowane w sposób optyczny. Wskazania nie powinny być maskowane przez żaden inny stan pracy urządzenia i powinny pozostawać aż do ręcznego skasowania.

13. Identyfikacja badanego weryfikowanego oprogramowania

W celu dokonania dokładnej identyfikacji weryfikowanego oprogramowania Producent powinien dostarczyć informacje w zakresie dokumentacji oprogramowania oraz nazwy plików stanowiących część oprogramowania ich wersji oraz rozmiaru ze wskazaniem daty ostatniej modyfikacji plików. Dane powinny zostać dostarczone w formie tabelarycznej jak przedstawiono w tabeli nr 1.

Tabela. 1

Oprogramowanie urządzenia dostarczonego do laboratorium				
Lp.	**Nazwa pliku**	**Wydanie / Wersja**	**Rozmiar (bajty)**	**Data**
1	np. Program ABC.c	np. V 1.0.0	np. 2 054	np. 01.01.2023
2	np. dev_module.h	np. V 2.3.1	np. 18 256	np. 01.01.2023
(…)	---	---	---	---
Dokumentacja oprogramowania dostarczonego do laboratorium				
Lp.	**Tytuł / Identyfikator**	**Opis (wersja, zawartość)**	**Rozmiar (bajty)**	**Data**
1	np. Dokument ABC	np. V 1.0.0	np. 325 206	np. 01.01.2023
2	np. html	np. 10 plików HTML	np. 65 203	np. 01.01.2023
3	np. png	np. 20 plików PNG	np. 923 460	np. 01.01.2023
(…)	---	---	---	---

Dodatkowo, Producent powinien zidentyfikować podstawowe dane techniczne w zakresie platformy sprzętowej, na której zainstalowane jest oprogramowanie. Dane powinny zostać dostarczone w formie tabelarycznej jak przedstawiono w tabeli nr 2.

Tabela. 2

Dane techniczne	
Typ procesora:	---
Typ płyty głównej:	---
Wersja oprogramowania:	---
Wersja sterowników:	---
Rodzaj pamięci:	np. FLASH / RAM / EEPROM