

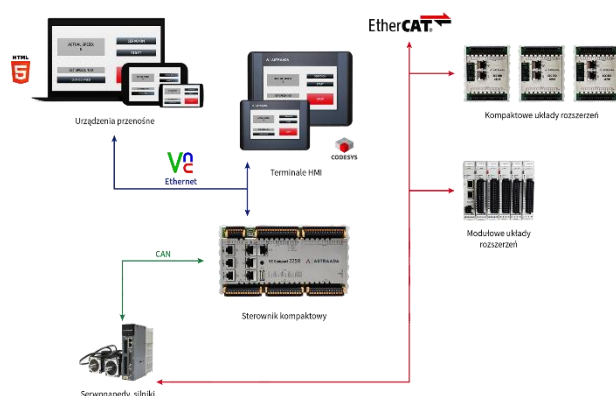
## ZDALNY DOSTĘP DO APLIKACJI W STEROWNIKACH ASTRAADA ONE Z WYKORZYSTANIEM SERWERA WWW

### Streszczenie

W artykule przedstawiono zagadnienia zdalnego podglądu i sterowania układów automatyki ze sterownikami PLC, za pośrednictwem przeglądarki internetowej. Testowano kompaktowe sterowniki Astraada One z wbudowanym serwerem sieci Web, przy tworzeniu aplikacji służących do zdalnego monitoringu rozproszonych systemów sterowania w ramach prowadzonych projektów dydaktycznych. Jedno środowisko, jedna baza zmiennych oraz symulator w pakiecie – CODESYS, jako wspólne środowisko obsługi programowej sterowników dają prawdziwy komfort ich programowania. Wskazano na bardzo ważną funkcjonalność sterowników Astraada One jaką jest wsparcie technologii Responsive Web Design – dopasowanie aplikacji do urządzeń mobilnych.

### WSTĘP

Programowalne sterowniki logiczne PLC z wbudowanym serwerem sieci Web pozwalają na stworzenie definiowanych przez użytkownika stron www oraz dostęp do CPU poprzez przeglądarkę sieciową na swoim komputerze lub urządzeniu mobilnym [1, 2]. Sterowniki z rodziny Astraada One to nowe, wyjątkowo wydajne urządzenia, działające w czasie rzeczywistym [3, 4], a dzięki zaimplementowaniu w nich takich protokołów komunikacyjnych jak m.in.: EtherCAT, BACnet, CANopen, Modbus TCP/IP idealnie sprawdzają się w pracy związanej z szeroko pojętą automatyką przemysłową (rys. 1). Urządzenia posiadają duże możliwości rozbudowy wejść i wyjść (zastosowanie dodatkowych modułów), a dzięki komunikacji sieciowej istnieje możliwość prowadzenia monitoringu i korzystania z wizualizacji poprzez stronę www. Sterowniki posiadają jedno wspólne dla wszystkich urządzeń środowisko programistyczne CODESYS zgodne z normą 61131-3. Ważną, bardzo elastyczną i wygodną własnością Astraada One jest również wsparcie technologii Responsive Web Design (RWD), która pozwala na idealne dopasowywanie zaprogramowanych aplikacji do urządzeń mobilnych. Dzięki tym rozwiązaniom PLC



Rys. 1. Produkty Astraada One oraz ich możliwości komunikacyjne (źródło: [5])

Astraada One mogą być sterowane przez laptopy, smartfony czy tablety z pominięciem specjalnych terminali HMI, co znacznie obniża koszty inwestycji, a jednocześnie pozwala sterować procesami z dowolnego miejsca na świecie z poziomu każdej przeglądarki internetowej wspierającej HTML5. Warto również podkreślić,

że zarówno wizualizacja jak też cała logika programu zaimplementowane są wewnątrz sterownika, dzięki czemu istnieje możliwość użycia jednej bazy zmiennych. Dzięki tej architekturze znacząco ułatwione jest programowanie sterowników oraz tworzenie wizualizacji, jak również jej konwersja na wersję webową.

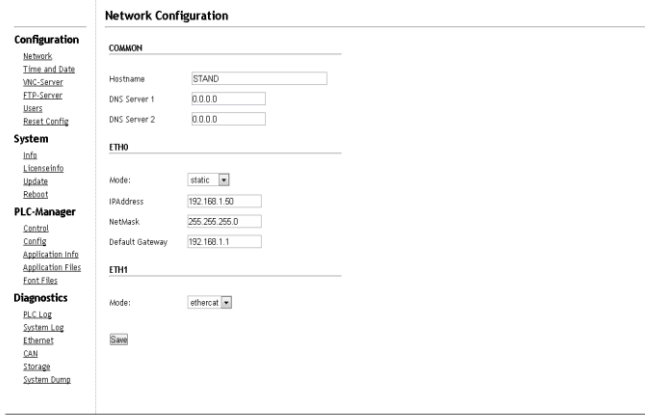
### 1. KONFIGURACJA URZĄDZENIA

Konfiguracja, sterowanie, zmiana adresu IP, czy też wymiana oprogramowania firmware w sterownikach Astraada One odbywa się z wykorzystaniem przeglądarki internetowej. Każdy ze sterowników posiada przypisany adres IP z puli: 169.254.255, gdzie dwie ostatnie cyfry adresu IP sterownika są to dwie ostatnie cyfry numeru seryjnego sterownika. Wyjątkiem będzie przypadek gdy numer seryjny sterownika kończy się na „00”. Wówczas jako ostatnie cyfry adresu IP zostaje przypisane „100”. Maską podsieci sterownika to 255.255.255.0. Gdy adres IP sterownika zostanie ustalony, należy przystąpić do skonfigurowania karty sieciowej. W tym celu otwieramy centrum sieci udostępniania systemu Windows. Następnie wybieramy opcję zmień ustawienia karty sieciowej, klikamy dwukrotnie na ikonę karty sieciowej do której podpięty jest nasz sterownik. Oznaczamy opcję protokół internetowy w wersji 6, pozostawiamy opcję protokół internetowy w wersji 4 – wchodzimy w jego właściwości, zaznaczamy opcję użyj następującego adresu IP, a w polu adres IP wpisujemy adres z rodziny naszego sterownika. W polu maska podsieci wpisujemy maskę urządzenia, zatwierdzamy i opuszczamy ustawienia. Po poprawnym skonfigurowaniu karty sieciowej, przechodzimy do otwarcia przeglądarki internetowej, a w pasku adresu wpisujemy adres IP naszego sterownika. W oknie przeglądarki wpisujemy Name: *admin*, Password: *admin*. Po wciśnięciu OK zostanie otwarte okno z dostępem do sterownika. Z bocznego menu mamy dostęp do wszystkich opcji i informacji na temat sterownika (rys. 2), poprzez podmenu Configuration, System, PLC-Manager oraz Diagnostics.

Podmenu Configuration. Zakładka Network – zmiany głównych ustawień sterownika: nazwa, IP, maska podsieci. Zakładka Time and date – zmiany godziny i daty. Zakładki VNC Server i FTP Server – ustawienia dostępu zdalnego. Zakładka Users – zmiany hasła użytkowników. Zakładka Reset Config – przywrócenia ustawień fabrycznych sterownika.



ECC2250 0.85 1131 - 25000100  
(plactive)



Rys. 2. Okno przeglądarki z dostępem do informacji i właściwości sterownika (źródło: dostęp DEMO do sterownika <http://195.205.24.110:5900>)

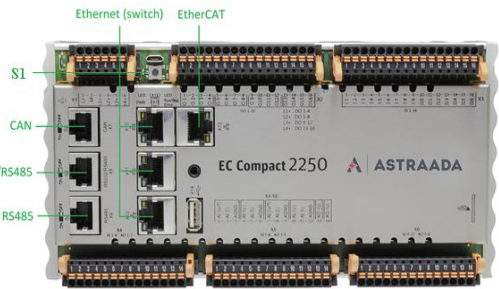
Podmenu System. Zakładka Info – informacja na temat systemu i sterownika: widać zainstalowane licencje, model urządzenia wraz z numerem seryjnym, temperaturę urządzenia, ilość pamięci całkowitej oraz wolnej. Zakładka Licence Info – informacja na temat licencji. Zakładka Update – możliwość uaktualnienia Firmware. Zakładka Reboot – ponowne uruchomienie systemu, konieczne by zapisać zmiany wprowadzone w konfiguracji sterownika.

Podmenu PLC Manager. Zakładka PLC Control – kontrola sterownika, uruchomienia i zatrzymania aplikacji, wyczyszczenia pamięci urządzenia. Zakładka Config – konfiguracja programu sterującego z karty SD. Zakładka Application Info – informacja na temat aplikacji, zakładka Application File – pliki wgrane do pamięci sterownika. Zakładka Font Files – pliki czcionek.

Podmenu Diagnostics. Zakładka PLC Log – rejestr zdarzeń sterownika. Zakładka System Log – dziennik zdarzeń systemu. Zakładki Ethernet oraz CAN – dane na temat sieci oraz podłączonych urządzeń. Storage – informacja o plikach systemowych. Zakładka System Dump – tworzenie pliku z zapisem z dziennika zdarzeń i rejestru systemu.

Po wprowadzeniu jakichkolwiek zmian należy zresetować sterownik poprzez zakładkę Reboot lub reset zasilania sterownika.

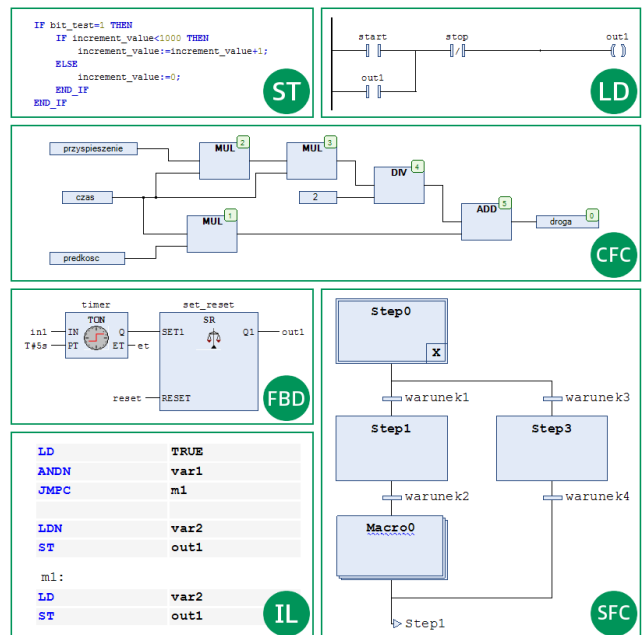
W przypadku, braku znajomości adresu sterownika istnieje możliwość przejścia w tzw. system Maintenance i zalogowania się przez domyślny adres IP. W tym celu należy wyłączyć zasilanie sterownika, nacisnąć i trzymać przycisk S1 w sterowniku, przywrócić zasilanie sterownika, trzymać wciśnięty przycisk S1 (rys. 3) dopóki nie zacznie migać dioda w kolorze żółto – czerwonym. Wówczas sterownik znajduje się we wspomnianym trybie Maintenance i można zalogować się przez domyślny adres IP. Przechodząc do zakładki Network możemy sprawdzić jaki adres jest ustawiony w sterowniku. Przejście w tryb Maintenance nie powoduje kasowania adresu IP i danych. Aby wyjść z tego trybu należy użyć funkcji Reboot lub zresetować zasilanie sterownika.



Rys. 3. Komunikacja w sterowniku kompaktowym Astraada One (źródło: [5])

## 2. CODESYS – ŚRODOWISKO PROGRAMOWANIA

Sterowniki Astraada One posiadają jedno wspólne środowisko w którym mamy możliwość zarówno programowania sterowników, tworzenia wizualizacji procesowych, jak również sterowania serwonapędami. Zmienne użyte w bazie nie wymagają duplikowania, gdy tworzymy program sterujący wraz z późniejszą wizualizacją. CODESYS V3.5 jest zgodny z normą 61131-3 i daje możliwość zaprogramowania sterownika we wszystkich językach znanych i wykorzystywanych w automatyce PLC: LD (schemat drabinkowy), FBD (funkcyjny schemat blokowy), ST (tekst strukturalny), IL (lista rozkazów), SFC – CFC (graf sekwencji) – (rys. 4).



Rys. 4. Języki programowania - CODESYS (źródło: [5])

W środowisku mamy również możliwość programowania obiektowego. Opcje podpowiadania składni, procesy upraszczające znajdowanie odpowiednich instrukcji oraz procedur, biblioteki z fragmentami gotowego kodu, a także nowoczesny debugger pozwalają, linijka po linijce, tworzyć i śledzić kod oraz zmienne w nim wykorzystane. Oprogramowanie CODESYS można bezpłatnie pobrać ze strony producenta i przetestować na zaimplementowanym symulatorze bez konieczności posiadania fizycznych urządzeń sterujących.

## 3. RESPONSIVE DESIGN W STEROWNIKACH ASTRAADA ONE

Panel operatorski posiadający funkcje zdalnego podglądu to do niedawna cecha występująca zwykle w sprzęcie z tzw. „górnej półki”. Nawet jednak, gdy panele HMI wyposażone są w Web Ser-

wer, to zaprogramowana w nich wizualizacja otwierana w przeglądarce jest statyczna, czyli posiada określone i stałe wymiary, zupełnie niezależne od rozdzielczości urządzenia na którym zostaje aktualnie wyświetlana. Problem pojawia się gdy aplikacja została zaprojektowana i napisana na bardzo duży panel operatorski, a chcemy ją wyświetlić na smartfonie czy tablecie. W sterownikach Astraada One istnieje możliwość automatycznego wykrycia wymiarów otwartego okna przeglądarki. Możemy więc zaprojektować kilka wizualizacji dla: dużego monitora czy też mniejszych wyświetlaczy w urządzeniach mobilnych. Takie rozwiązanie w procesie projektowania wizualizacji nosi nazwę Responsive Web Design (rys. 5).



Rys. 5. Responsive Web Design w Astraada One (źródło: [6])

Aby zaprojektować wizualizację wykorzystując technologię Responsive Web Design nie ma potrzeby stosowania wielu problemowych i trudnych operacji. Projektowanie jest uproszczone w maksymalny sposób i sprowadza się do określenia wymiarów urządzeń dla których wizualizacje te będą wyświetlane oraz do zapro-

nazwę ekranu który jest w tym momencie wyświetlany, a także sprawdza wymiary okna przeglądarki. Wszystkie te dane wraz ze wskaźnikiem osoby aktualnie wyświetlającej wizualizację zostają przekazane do funkcji SetVisualization.

Za szerokość oraz wysokość okna przeglądarki odpowiadają zmienne iWidth oraz iHeight wyrażone w pikselach. Odpowiednia funkcja dokonuje porównania wymiarów z tymi wartościami które ustawił programista, aby na tej podstawie stwierdzić rodzaj urządzenia. Coraz częściej mamy do czynienia z faktem, iż nowe smartfony posiadają większą rozdzielczość od starszych tabletów. Aby rozwiązać taki problem stosujemy porównywanie stosunku wysokości do szerokości, z uwagi na fakt, iż ekrany smartfonów posiadają zwykle inne proporcje od ekranów tabletów.

Gdy funkcja rozpozna typ urządzenia następuje dodanie do zmiennej sCurrentVisuName, czyli głównego członu nazwy ekranu, właściwego przyrostka i następuje przełączenie wizualizacji.

Liczbę liter przyrostka, który określa typ urządzenia należy podać w programie VisuClientHandler, a informacja ta podana wraz z wartością przechowywana jest w zmiennej iSufixLength (rys. 6).

W funkcji SetVisualization określamy jakie są warunki przełączania wizualizacji. W poniższej, przykładowej aplikacji funkcja rozpoznaje jedynie czy ekran urządzenia ustawiony jest pionowo, czy poziomo. Odpowiedni przyrostek zostaje dodany do nazwy ekranu i wywoływana jest funkcja SetMainVisu:

```
IF iWidth > iHeight THEN
    VisuElems.g_VisuManager.SetMainVisu(pClient,    CON-
CAT(sCurrentVisuName, 'H'));

```

| Scope | Name         | Address          | Data type                                 | Initialization | Comment | Attributes |
|-------|--------------|------------------|---|----------------|---------|------------|
| 1     | VAR          | pClient          | POINTER TO VisuElems.VisuStructClientData |                |         |            |
| 2     | VAR          | tmr              | TON                                       |                |         |            |
| 3     | VAR          | iWidth           | INT                                       |                |         |            |
| 4     | VAR          | iHeight          | INT                                       |                |         |            |
| 5     | VAR          | sCurrentVisuName | STRING                                    |                |         |            |
| 6     | VAR          | itfVisu          | VisuElems.VisuElemBase.IVvisualisation    |                |         |            |
| 7     | VAR CONSTANT | iSufixLength     | INT                                       | 1              |         |            |

Rys. 6. Okno programu VisuClientHandler (źródło: [6])

jektowania ekranów graficznych, dla każdego z urządzeń. Wykonanie własnej aplikacji wymaga:

- sterownika Astraada One z serii ECC2000 z wersją firmware 1.7.1 lub nowszą;
- środowiska programowania CODESYS V3.5 z Service Pack odpowiednim dla posiadanej wersji firmware sterownika (dla firmware w wersji 1.7.1 jest to SP7).

Aby wizualizacja działała w sposób poprawny, należy używać schematu, polegającego na jednolitym nazewnictwie tworzonych wizualizacji. Podstawowy człon nazwy winien być jednakowy dla każdego typu urządzenia, a dla rozróżnienia zawierać przyrostek określający konkretny typ. Dla każdego typu urządzenia określamy odpowiednią (taką samą) ilość liter. Na przykład jeżeli główny ekran naszej wizualizacji nazwiemy MainScreen to odpowiednimi nazwami dla przyrostków określających mogłoby być: TablVer, SmfnHor, Monitor czy TablHor lub SmfnVer.

Za poprawne działanie Responsive Web Design odpowiadają dwa POU (ang. Program Organization Unit): program VisuClientHandler oraz funkcja SetVisualization. VisuClientHandler należy wywołać w zadaniu VISU\_TASK. Program w sposób cykliczny sprawdza każde urządzenie wyświetlające wizualizację, odczytuje

```
ELSE
    VisuElems.g_VisuManager.SetMainVisu(pClient,    CON-
CAT(sCurrentVisuName, 'V'));

```

```
END_IF

```

- pClient - jest wskaźnikiem przyporządkowanym do konkretnego klienta wyświetlającego aktualnie wizualizację.
- sCurrentVisuName - jest głównym członem nazwy wyświetlanego ekranu.
- 'H' i 'V' - to przyrostki które określają orientację urządzenia.

Jeżeli chcemy przetestować działanie aplikacji możemy otworzyć wizualizację w przeglądarce komputera, następnie zmieniać rozmiar okna. Gdy odświeżymy stronę, wizualizacja będzie dostosowywana do rozmiarów okna.

Warunki w funkcji SetVisualization, które sprawdzają typ urządzenia, czy to poprzez stosunek jego szerokości i wysokości, czy też bezpośrednio poprzez wymiary okna przeglądarki, należy dobrać w taki sposób, aby odpowiadały faktycznym urządzeniom, na których będzie oglądana wizualizacja. Jeśli przejdziemy w tryb online możemy łatwo sprawdzić wymiary okna przeglądarki ostat-



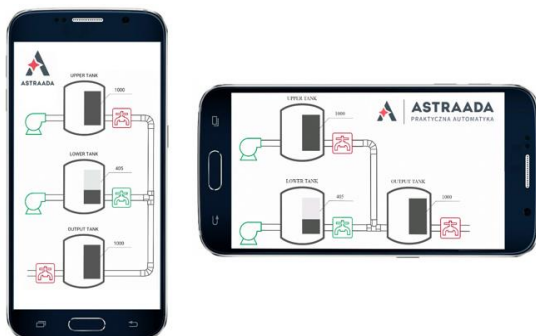
niego połączonego klienta, które są przechowywane w zmiennych iWidth oraz iHeight programu VisuClientHandler (rys. 7).

| Device.Application.VisuClientHandler |                       |              |
|--------------------------------------|-----------------------|--------------|
| Expression                           | Type                  | Value        |
| pClient                              | POINTER TO VisuEle... | 16#00000000  |
| tmr                                  | TON                   |              |
| iWidth                               | INT                   | 770          |
| iHeight                              | INT                   | 471          |
| sCurrentVisuName                     | STRING                | 'MainScreen' |
| itfVisu                              | IVisualisation        | 16#03AA7114  |
| iSufixLength                         | INT                   | 1            |

Rys. 7. Okno programu VisuClientHandler (źródło: [6])

Dla uzyskania większego komfortu obsługi jak również znacznej uniwersalności w poprawnym rozpoznawaniu typu urządzenia możemy wprowadzić uproszczenie zakładające, że wizualizacja na smartfonach oglądana będzie zawsze w pozycji pionowej, natomiast na tabletach zawsze w pozycji poziomej. Przy tym założeniu należy jedynie sprawdzić, czy wysokość okna przeglądarki jest większa od szerokości - pozycja pionowa - czy też odwrotnie - pozycja pozioma.

Aby zweryfikować działanie przykładowej aplikacji z wykorzystaniem RWD, możemy skorzystać ze strony: <http://195.205.24.110:502/webvisu.htm> (rys. 8). Interfejs webowy oferowany przez producenta daje możliwość stworzenia do 100 jednoczesnych połączeń i pozwala na wgląd online oraz kontrolowanie danego procesu.



Rys. 8. Przykładowa aplikacja z wykorzystaniem RWD (źródło: dostęp DEMO <http://195.205.24.110:502/webvisu.htm>)

## PODSUMOWANIE

Sterowniki Astraada One znajdują zastosowanie w coraz bardziej zróżnicowanych dziedzinach. Dzięki łatwości programowania, tworzenia wizualizacji, jak również łatwości zdalnego dostępu do zaprogramowanych procesów wykorzystywane są w coraz bardziej skomplikowanych procesach. Bieżące śledzenie wprowadzonych nastaw daje możliwość pełnej kontroli, zwłaszcza, że może być realizowane zdalnie przez wiele osób jednocześnie. Do całościowego monitoringu działania systemu wystarczy urządzenie mobilne z dostępem do sieci internet, oraz dowolna przeglądarka, np. Firefox lub Chrome. Mamy możliwość wykonywania do 100 równoległych połączeń.

Programowalne sterowniki logiczne PLC, wyposażone we wbudowany Web Serwer, znajdują coraz szersze zastosowanie w procesach takich jak:

- integracja rozproszonych systemów diagnostyki i sterowania (np.: transport, przemysł przetwórczy czy wydobywczy);

- rozwój automatyki budynkowej BMS (np.: sterowanie oświetleniem, systemy HVAC - ogrzewanie, wentylacja i klimatyzacja, kontrola zużycia energii czy ochrona i bezpieczeństwo);
- produkcja na rzecz OEM (np.: pojazdy i urządzenia mobilne, maszyny do pakowania i nadruku, myjnie samochodowe lub urządzenia obróbcze).

Omówione właściwości sterowników PLC Astraada One stanowią punkt wyjścia do badań w zakresie programowania i wizualizacji procesów w środowisku CODESYS przy wykorzystaniu technologii Responsive Web Design. Pozwoli to na rezygnację z terminali HMI na rzecz urządzeń mobilnych, przy tworzeniu aplikacji służących do zdalnego monitoringu rozproszonych systemów sterowania w ramach prowadzonych przez autorów projektów przemysłowych i dydaktycznych.

## BIBLIOGRAFIA

1. *Simatic S7-1200 Easy book Manual 03/14* Siemens, Warszawa 2014
2. *Astraada One – nowa linia kompaktowych i modułowych sterowników w ofercie ASTOR*, Biuletyn Automatyki Astor Nr 85, 3/2015
3. *Sterownik PLC Astraada One produktem roku 2015/2016 czasopisma Napędy i Sterowanie*, Biuletyn Automatyki Astor Nr 87, 1/2016
4. *Astraada One – nowa linia kompaktowych i modułowych sterowników w ofercie ASTOR*, Napędy i sterowanie, Nr 2/2016
5. *Dokumentacja Techniczna - Astraada,- Sterowniki PLC*, Astor, <http://www.astor.com.pl/wsparcie/dokumentacja-techniczna/Astraada/Sterowniki%20PLC/Astraada%20One%20ECC2000/>
6. *Informator Techniczny firmy Astor*, 2016

## REMOTE ACCESS TO APPLICATIONS IN ASTRAADA ONE CONTROLLERS WITH THE USE OF THE WEB SERVER

### Abstract

*This paper presents issues related to remote monitoring and control of automation systems with PLCs, via a web browser. The compact cocontrollers Astraada One with built-in Web Server was tested for creating applications for remote monitoring of distributed control systems within the framework of conducted didactic projects. One environment, one variable base, and a simulator in a package - CODESYS, as a common controller programming environment, give them real programming comfort. The very important functionality of the Astraada One controllers, which is the support of Responsive Web Design technology, matching applications to mobile devices, was pointed out.*

Autorzy:

**Mariusz Zabroń**, student – UTH Radom – Wydział Informatyki i Matematyki

Dr hab. inż. **Andrzej Puchalski**, prof. nadzw. UTH Radom – Wydział Mechaniczny, Instytut Eksploatacji Pojazdów i Maszyn.