

Janusz SZPYTKO ¹
Paweł HYLA ²

ROLA I FUNKCJONALNOŚĆ WBUDOWANYCH ŚRODOWISK OPERACYJNYCH W SYSTEMACH WIZYJNYCH

Popularność wbudowanych systemów operacyjnych jest wyznacznikiem postępu technologicznego i znakiem współczesnych czasów. Systemy wbudowane umożliwiają implementację cech inteligencji (ang. smart feature) zarówno do urządzeń specjalizowanych jak i codziennego użytku. Główną cechą urządzeń inteligentnych stanowi autonomiczność w wykonywaniu powierzonych działań.

Wypowiedź koncentruje uwagę na roli i funkcjonalności wbudowanych systemów operacyjnych (ang. embedded) dla procesów nadzorowania wizyjnego. W artykule przedstawiono i porównano systemy wbudowane skompilowane na bazie rodziny systemów Windows oraz Unix. Porównanie systemów wbudowanych zostało przeprowadzone na podstawie otrzymanych funkcjonalności systemu wizyjnego z zastosowaniem inteligentnej kamery.

EMBEDDED SOFTWARE FUNCTIONALITY IN THE AUTOMATIC IDENTIFICATION PROCESS ON THE VISION SYSTEMS BASE

At present time embedded software is a sign of time. This type of software enable intelligence implementation feature (so called smart feature) to the variety kind of the devices. The devices enrich by the embedded software systems are more effectiveness. Especially vision methods in automatic identification processes are determining by the embedded software functionality. The software linking hardware possibility with the methods for example tag scanning.

The main target of statement is role and the functionality of embedded system software compiled on MS Windows and Unix platform, dedicated for machine vision support and automatic identification method.

¹Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Inżynierii Mechanicznej i Robotyki; 30-059 Kraków; al. A. Mickiewicza 30,

Tel: +48 12 617 3103, Fax: +48 12 617 3531, e-mail: szpytko@agh.edu.pl

²Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie, Wydział Inżynierii Mechanicznej i Robotyki; 30-059 Kraków; al. A. Mickiewicza 30, Tel: +48 12 617 3104, e-mail: hyla@agh.edu.pl

1. WSTĘP

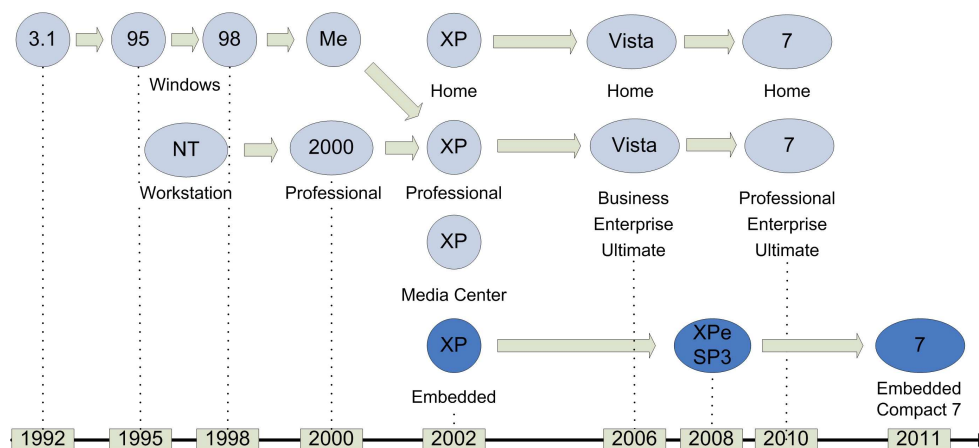
Innowacyjność współczesnego urządzenia technicznego determinuje multi-dyscyplinarny charakter podzespołów użytych w ramach architektury układów mechatronicznych [15], stanowiących kombinację podzespołów mechanicznych, elektronicznych oraz układów automatycznego sterowania [4]. Termin mechatronika pojawił się po raz pierwszy w roku 1969 w Japonii. Zgodnie z definicją przyjętą przez International Federation for the Theory of Machines and Mechanism, mechatronika jest *synergiczną kombinacją mechaniki precyzyjnej, elektronicznego sterowania i systemowego myślenia przy projektowaniu produktów i procesów produkcyjnych*.

Potrzeby programowania urządzeń mechatronicznych polegający na adaptacji interfejsu urządzenia pod potrzeby i wymagania konkretnego użytkownika poskutkowało powstaniem interfejsów semantycznych [14] z nową koncepcją architektury oprogramowania, tak zwanych systemów wbudowanych.

Artykuł jest zorientowany na możliwości kształtowania urządzenia typu *smart* [3, 16] poprzez funkcjonalność wbudowanego systemu operacyjnego na przykładzie architektury systemu wizyjnego [17].

2. WBUDOWANE SYSTEMY OPERACYJNE

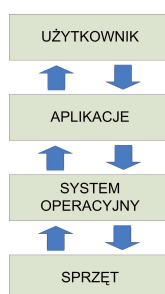
Systemy wbudowane cechuje charakterystyczny dobór elementów składowych, polegający na ukierunkowanej obsłudze ściśle określonych komponentów sprzętowych. Im mniej złożone i bardziej wyspecjalizowane jest oprogramowanie jakiegokolwiek urządzenia, tym poziom niezawodności całego systemu (ang.: *hardware, software*) rośnie. System wbudowany (ang. *embedded system*) [5], to system operacyjny specjalnego przeznaczenia dedykowany dla specyficznego środowiska sprzętowego. Komercyjne systemy wbudowane wykształciły się w wyniku ewolucji systemów operacyjnych (rys. 1.). System wbudowany wyodrębniony na bazie systemu operacyjnego XP miał swą premierę w 2002 roku.



Rys. 1. Ewolucja systemów operacyjnych z rodziny Windows i Workstation

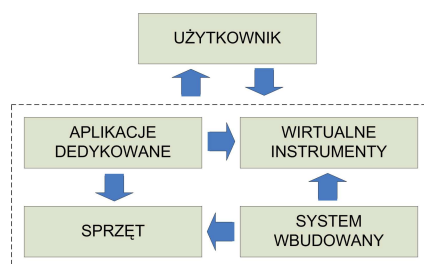
Cechę charakterystyczną systemów wbudowanych stanowi unikalna kompozycja zachowująca jądro (ang. *kernel*), powłokę (ang. *interface*) oraz system plików (ang. *file system*) systemu macierzystego z kompozycją sterowników przeznaczonych wyłącznie dla struktury komponentów konkretnego urządzenia [8]. Poprzez unikatową kompozycję dopasowaną tylko i wyłącznie do danej sprzętowej specyfikacji urządzenia, system wbudowany staje się jego integralną częścią.

Tradycyjnie środowisko zarządzające sprzętem komputerowym (ang. *OS Operating System*), posiada architekturę otwartą, umożliwiającą dodawanie oraz instalowanie nowego sprzętu, a środowisko do uruchamiania i kontroli zadań, komunikuje się z użytkownikiem poprzez aplikacje tworzące interfejs urządzeń *wejście/ wyjście* (rys. 2).



Rys. 2. Schemat relacji logicznych pomiędzy elementami systemu komputerowego typu OS

W systemach wbudowanych, system operacyjny jest najczęściej trwale zintegrowany z aplikacjami, tworząc jednorodny interfejs typu HMI (ang. *Human Machine Interface*) [9], będący modułem bezpośredniej komunikacji pomiędzy człowiekiem, a urządzeniem (rys. 3).



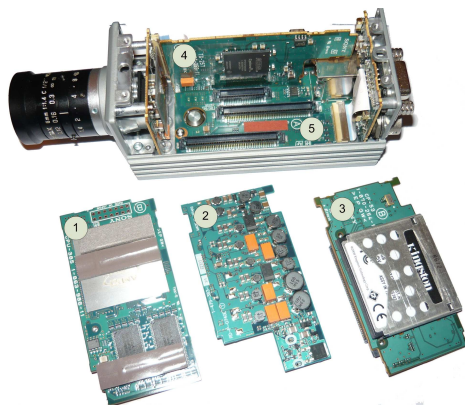
Rys. 3. Schemat relacji logicznych pomiędzy elementami systemu komputerowego typu Embedded

Kompozycja sterowników systemu wbudowanego nie pozwala na dodawanie nowych urządzeń (chyba że zostanie to przewidziane w fazie kompilacji), a rolę urządzeń *wejście/ wyjście* przejmują najczęściej wirtualne instrumenty, typu wirtualna klawiatura lub ekrany dotykowe.

2.1. Architektura urządzeń inteligentnych

Urządzenia inteligentne łączą w sobie funkcję pozyskiwania informacji wraz z jej jednoczesnym przetwarzaniem. Nierzadko posiadają także umiejętność podejmowania autonomicznej decyzji w oparciu o posiadane predyktywne lub adaptacyjne wzorce zachowań istniejące w bazie danych. W urządzeniach typu inteligentnego autonomiczna struktura powstaje w rezultacie łączenia różnorodnych układów i wykorzystywania efektu synergii ich wzajemnego oddziaływania wraz z algorytmami implementującymi sztuczną inteligencję. Algorytmy heurystyczne, sieci neuronowe, systemy ekspertowe oraz logika rozmyta umożliwiają przetwarzanie informacji w wiedzę w fazie projektowania (sposób pośredni) lub eksploatacji (sposób bezpośredni) poprzez *obserwację* rzeczywistości i wykorzystywanie analogii. Systemy wbudowane poprzez zwartą architekturę umożliwiły ekspansję urządzeń mobilnych oraz implementację cech właściwych dla *inteligencji* do różnego typu urządzeń. Dodatkowo, sukces systemów wbudowanych to także urządzenia wirtualne (ang. *virtual instruments*) w których oprogramowanie, jego funkcjonalność oraz architektura odgrywają rolę decydującą. Dodatkową zaletę stanowi eliminacja zawodnych podsystemów mechanicznych charakteryzujących się bezwładnością.

Architektura kamery, jako urządzenia inteligentnego (rys. 4) jest prosta do zrealizowania. Cyfrowy układ wizyjny w sposób naturalny łączy elementy pozyskiwania informacji wykorzystujący matryce wykonane w technologii CCD/ CMOS (ang. *Charge-Coupled Device/ Complementary Metal–Oxide–Semiconductor*), układy transmisji informacji oraz zautomatyzowane układy wykonawcze zbudowane na elektronicznej migawce (ang. *trigger*). Inteligentna kamera (ang. *smart cameras*) [10, 16] w systemie nadzorowania wizyjnego znajduje zastosowanie w odczycie i weryfikacji napisów, dekodowaniu zakodowanych informacji (kody dwuwymiarowe), pomiaru wielkości geometrycznych, identyfikacji obecności obiektów na obrazie oraz kontroli ich kształtu. Rola jaką spełnia inteligentna kamera w systemie wizyjnym jest przede wszystkim predefiniowana zainstalowanym oprogramowaniem, jego funkcjonalnością i rozwiązaniami czysto programowymi zaimplementowanymi w system wbudowany zarządzający pracą kamery.



Rys. 4. Architektura inteligentnej kamery o oznaczeniu kodowym XCI-V3

Architektura sprzętowa kamery Xci-V3 posiada modułową budowę podzespołów osadzoną na *platce głównej* (5) – rys.4. Poszczególne karty rozszerzeń zawierają jednostkę obliczeniową (1), układy pamięci RAM (ang. *Random-Access Memory*) oraz ROM (ang. *Read Only Memory*) (2), magazyn pamięci wymiennej w skład którego wchodzi slot karty CF (ang. *Compact Flash*) (3) oraz układ przetwarzania sygnału matrycy światłoczułej (4). Karta sieciowa, kontroler USB oraz układ graficzny jest zintegrowany wraz z *platą główną* (5), zgodnie ze współczesnym standardem architektury połączeń pomiędzy pamięcią, procesorem i urządzeniami *wejście/ wyjście*.

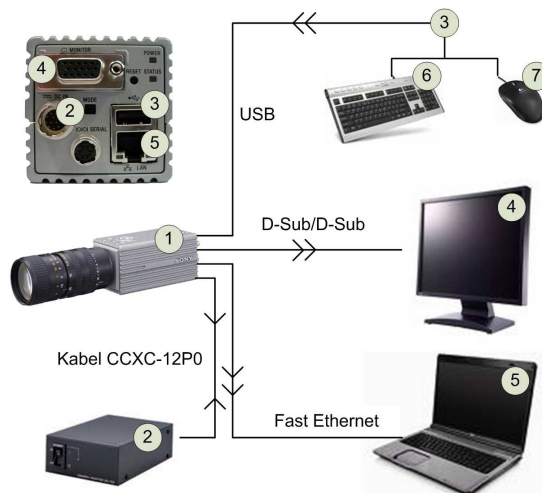
2.2. Analiza obrazu i kontrola wizyjna wykorzystujące urządzenie typu inteligentnego

Do testów funkcjonalności i możliwości oprogramowania wbudowanego została użyta inteligentna kamera o oznaczeniu kodowym: XCI V3 z matrycą 1/3'' [11, 12, 13], wykonana w technologii CCD i progresywnym skanowaniem firmy Sony. Parametry techniczne przedstawiono w Tab.1. Jako układ optyczny zastosowano obiektyw ze stałą ogniskową 6 mm firmy Pentax (Tab. 1.).

Tab. 1. Parametry systemu wizyjnego

kamera:	XCI V3	obiektyw:	H614-M
	SENSOR CCD typ: monochromatyczny 1/3'' rozdzielczość efektywna: 640x480 VGA PROCESOR typ: x86 AMD Geode GX533, 400 MHz PAMIĘĆ OPERACYJNA pamięć wew.: 256 MB DDR-SDRAM pamięć wymienna: Compact Flash INTERFACE ethernet: 100 Base-Tx/10 Base-USB: 1.1 monitor out: D-sub 15 pin serial interface: RS232C INNE mocowanie obiektywu: C-mount odporność na przeciążenia: 70 G odporność na wstrząsy: 10 G		OPTYKA ogniskowa: 6mm (+/-5 %) wskaźnik przysłony: 1:1.14 przysłona: F/1.4 – F/16 kąt widzenia: <i>ukośny</i> : 69,0° <i>poziomy</i> : 57,4° <i> pionowy</i> : 44,4° makro: 0,2 m

Na rys. 5, przedstawiono schemat połączeń systemu wizyjnego typu *machine vision* [17] dla potrzeb testu funkcjonalności systemów wbudowanych z rodziny Unix oraz Windows zarządzających pracą inteligentnej kamery Sony XCI V3. Sygnał wizyjny rejestrowany przez kamerę w postaci pojedynczego zdjęcia, podlega akwizycji oraz analizie obrazu. Kamera została wyposażona w dodatkowe interfejsy *wejście/ wyjście* oraz stację nadzorowania dla potrzeb zdalnego nadzoru i monitoringu.



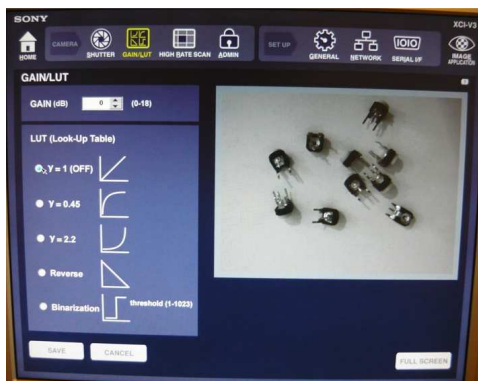
Rys. 5. Architektura systemu wizyjnego nadzorowania z użyciem inteligentnej kamery:
 1 – kamera XCI V3, 2 – stacja zasilająca DC-700CE, 3- koncentrator USB,
 4 – interfejs wyjścia, 5- host, 6,7 – interfejsy wejścia

3. OPROGRAMOWANIE WBUDOWANE TYPU UNIX

Systemy wbudowane powstają także na bazie wolnego i otwartego oprogramowania typu FLOSS (ang. *Free Libre/ Open Source Software*). Dla potrzeb urządzeń inteligentnych na bazie jądra Linuxa zostało zaprojektowane środowisko cyfrowe MontaVista Linux. Odpowiednio dostosowane dystrybucje znalazły zastosowanie w urządzeniach inteligentnych przeznaczonych docelowo dla domu biura: systemy typu SoHo (ang. *Small Home Office*), urządzeń stosowanych w medycynie: systemy typu EfM (ang. *Electronic for Medicine*) oraz panele z softwarowymi kontrolkami typu VIP (ang. *Virtual Instruments Panel*).

Wersja systemu z oznaczeniem „Pro” (ang. *Professional*) zawierająca nakładkę z interfejsem GUI (ang. *Graphic User Interface*) została zaimplementowana w kamerę Sony typu XCI V3. Zaimplementowane oprogramowanie umożliwia przeprowadzenie podstawowej inspekcji i analizy obrazu w kilku trybach:

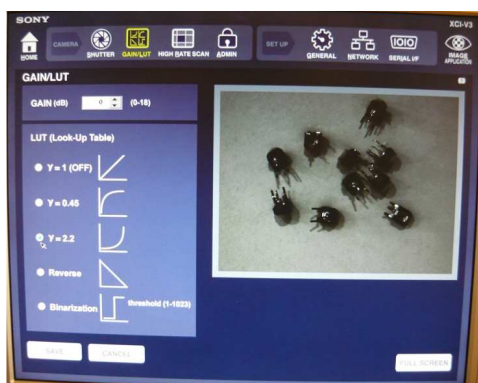
- modulacji gamma usuwającą z obrazu monochromatycznego zniekształcenia poprzez redukcję kontrastu obrazu wejściowego (rys. 6-8),
- pozyskania z obrazu przeciwieństwa jasności tonów z uwzględnieniem pełnej skali dynamiki pozyskanego materiału (rys. 9),
- binaryzacji w wyniku której ilość informacji na temat każdego punktu obrazu zostaje ograniczona do dwóch wartości (0, 1): oprogramowanie umożliwia realizowanie progów binaryzacji w zakresie 1-1023 (rys. 10).



Rys. 6. Obraz 1:1 (bez modulacji; $\gamma=1$)



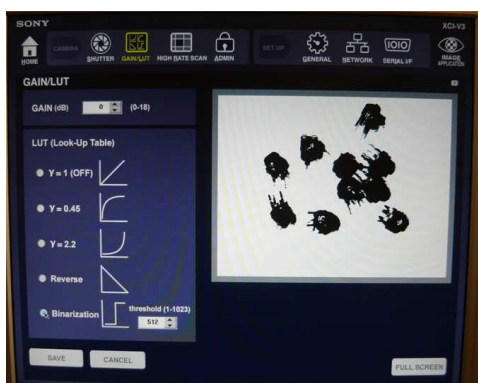
Rys. 7. Obraz z modulacją; $\gamma=0,45$



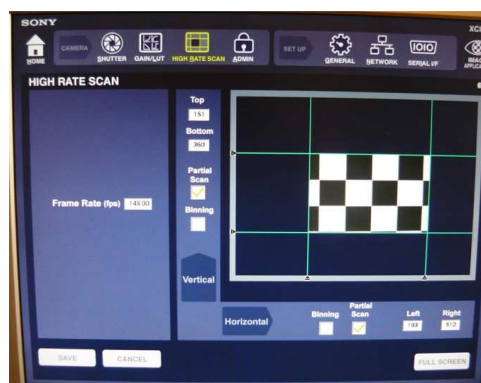
Rys. 8. Obraz z modulacją; $\gamma=2,2$



Rys. 9. Negatyw zdjęcia



Rys. 10. Binaryzacja obrazu



Rys. 11. Skanowanie fragmentaryczne

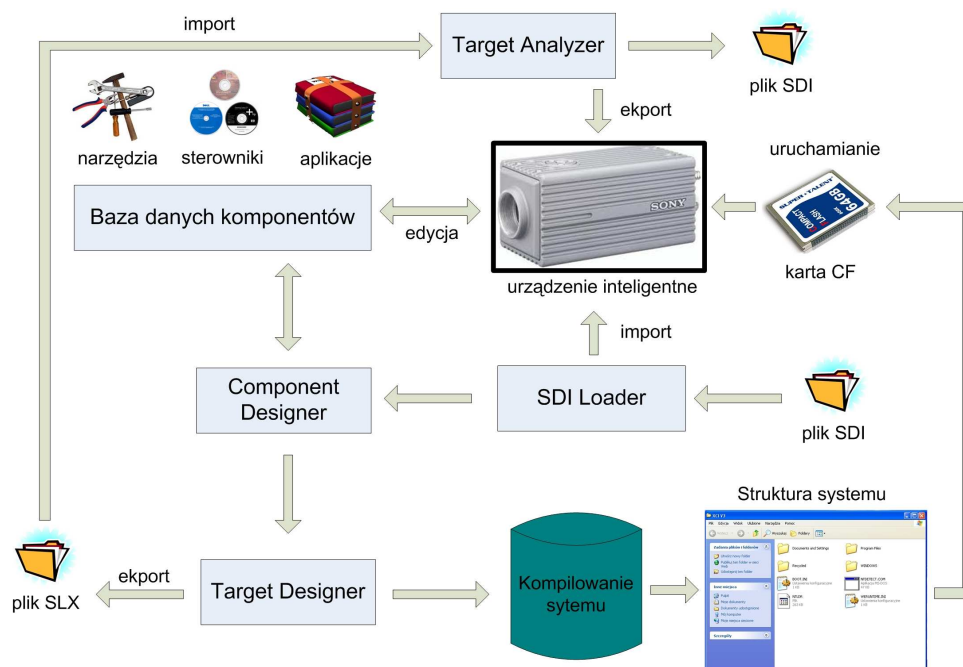
Oprogramowanie wbudowane MontaVista Linux, posiada także funkcję softwarowej fragmentaryzacji obrazu, pozwalającą wyeksponować w rejestrowanym obrazie jedynie interesujący obszar zdjęcia (rys. 11). W trybie fragmentaryzacji, dodatkowe okno dialogowe umożliwia ustawienie szybkości rejestracji obrazu, poprzez ustawienie tak zwanego *frame ratio*. Podgląd obrazu rejestrowanego przez kamerę może być udostępniony w sieci LAN, po ustawieniu w opcji *General*, funkcji *Camera Image* na wartość *On*. Jest to bardzo proste rozwiązanie, ale bardzo istotne ze względu na brak możliwości zapisywania na pamięci wewnętrznej kamery zrzutów zarejestrowanych obrazów.

Główną zaletę wbudowanego Linuxa, stanowi szybkość działania zaimplementowanych aplikacji, intuicyjność obsługi i prostotą interfejsu GUI. Dodatkowo interfejs GUI obciąża zasoby systemowe jedynie w kilku procentach, co wpływa na niezawodność i stabilność pracy kamery, poprzez niską emisję ciepła (układ chłodzenia procesora Geode jest oparty na pasywnych elementach) i stabilną temperaturę pracy. Aplikacja nadzorująca układy kamery, udostępnia wirtualną klawiaturę ekranową umożliwiającą wprowadzanie danych alfanumerycznych przy użyciu urządzenia wskazującego.

Znikoma ilość zaimplementowanych narzędzi przeznaczonych bezpośrednio do analizy obrazu, nie pozwala w pełni wykorzystać możliwości hardwarowych kamery XCI V3. W szczególności brak możliwości wprowadzenia do pamięci wewnętrznej kamery wzorca kontrolnego w postaci zdjęcia bądź jego wydzielonego fragmentu wpływa na brak możliwości klasyfikowania elementów i wykrywania defektów. Braki w oprogramowaniu w zakresie omawianej funkcjonalności eliminują zastosowanie systemu wbudowanego typu MontaVista Linux, dla profesjonalnego zastosowania w systemach widzenia maszynowego w wizyjnej inspekcji zautomatyzowanych ciągów transportowo-technologicznych [6].

4. OPROGRAMOWANIE WBUDOWANE XPe

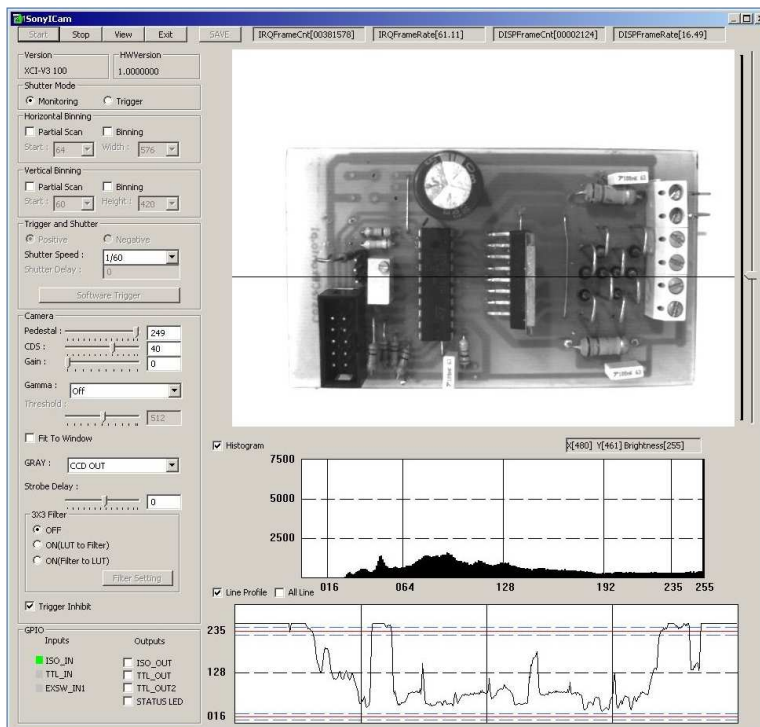
System wbudowany w wersji Windows XP Embedded (XPe) jest *uszczuploną* wersją systemu operacyjnego Windows XP. Architektura systemu zakłada, że każda wersja systemu jest dostosowywana do potrzeb danej specyfiki sprzętu. Do celu przygotowania systemu dla urządzeń inteligentnych, zostało oddane użytkownikom dodatkowe środowisko Embedded Studio (rys. 12). Embedded Studio [2], zawiera szereg aplikacji niezbędnych dla potrzeb projektowania systemu wbudowanego pod względem architektury systemu wbudowanego (*Target Analyzer*), dobierania poszczególnych komponentów (*Component Designer*), użycia gotowych plików SDI z kompozycją sterowników (*SDI Loader*) oraz kreatora systemu wbudowanego (*Target Designer*). Proces kompilacji systemu wbudowanego opartego o platformę Windows jest czasochłonny, skomplikowany i wymaga dużej wiedzy na temat architektury systemów operacyjnych, architektury sprzętu dla którego system jest dedykowany oraz funkcji realizowanych przez urządzenie inteligentne w celu dobrania i zintegrowania dodatkowych aplikacji wspomagających.



Rys. 12. Schemat blokowy przygotowania kompilacji systemu XPe w Embedded Studio

Do każdego procesu kompilacji jest załączony raport zawierający potencjalne błędy. Po uzyskaniu bezbłędnego raportu istnieje możliwość przeniesienia obrazu systemu na zewnętrzny nośnik. Po pierwszym uruchomieniu w urządzeniu docelowym, użytkownik ma do dyspozycji w pełni funkcjonalne środowisko operacyjne, na pierwsze spojrzenie przypominające tradycyjnego Windowsa XP. Zasadnicza różnica funkcjonowania systemu XPe w porównaniu z systemem operacyjnym XP, to brak istnienia pliku wymiany danych oraz brak przydzielenia pamięci wirtualnej. Zabiegi te mają na celu wyeliminowanie „operacji dyskowych” wykonywanych przez system, w celu wydłużenia żywotności karty pamięci przechowującej dane.

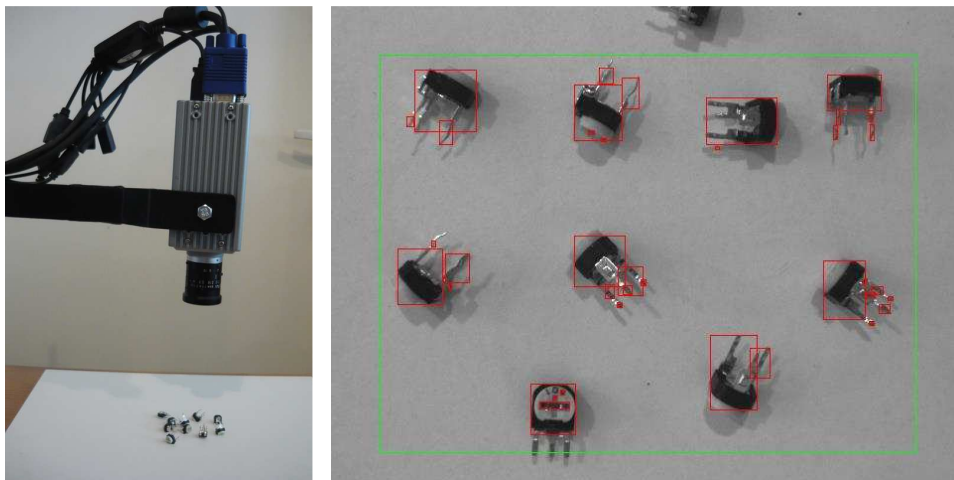
Dodatkowo z systemem została sprzęgnięta aplikacja *SonyiCam* (rys. 13), dostarczona przez producenta w pliku SDI, posiadająca sprzęgnięte narzędzia do analizy obrazu podobne do interfejsu MotaVista Linux. Jedyną różnicą polega na możliwości podglądu *on-line* w przebieg histogramu całego rejestrowanego obrazu lub pojedynczej linii z zakresu rozdzielczości zdjęcia (1-480). Softwarowa nakładka na system wbudowany umożliwia również fragmentaryzację obrazu oraz czas otwarcia programowej migawki.



Rys. 13. Analiza obrazu w aplikacji SonyICam

3.2.1. Aplikacje wspomagające

Niewątpliwą zaletą systemu wbudowanego XPe, stanowi możliwość instalowania aplikacji wspomagających od innych producentów, związanych z analizą obrazu. Przykładem aplikacji wspomagającej jest oprogramowanie firmy National Instruments, dedykowane dla potrzeb inteligentnych kamer: *Vision Builder for Automated Inspection* [7]. Oprogramowanie *Vision Builder* w pełni współpracuje ze środowiskiem NI LabView, ale stanowi samodzielny produkt. Po zainstalowaniu środowiska istnieje możliwość uruchomienia programu bezpośrednio na kamerze i używania jej interfejsu do nadzorowania programu lub użycia interfejsu sieciowego z uruchomionym VB na komputerze typu host z podłączoną inteligentną kamerą kablem sieciowym. Szereg funkcjonalności jakie dostarcza instalacja VB, to m.in.: możliwość wykrywania zdefiniowanych elementów na pozyskanym obrazie wraz z jednoczesnym ich zliczaniem (rys. 14).

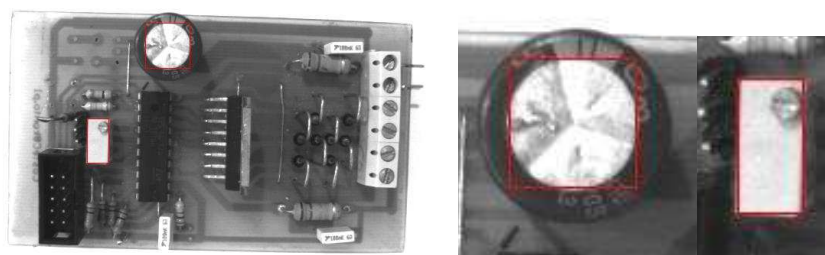


Rys. 14. Przykładowe stanowisko inspekcji podzespołów elektronicznych

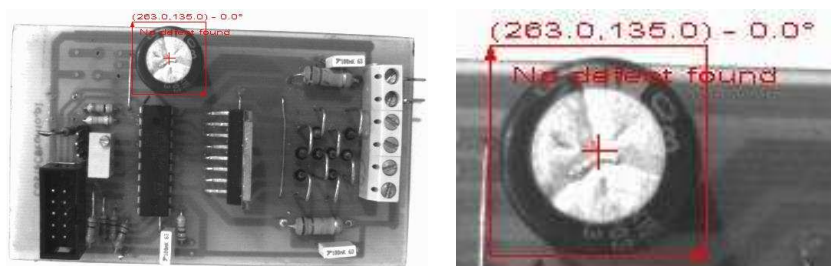
Vision Builder umożliwia uruchamianie programów napisanych w języku programowania obiektowego dedykowanych dla celów inspekcji. Vision Builder zawiera ponad sto funkcji, filtrów oraz operacji logicznych pozwalających opracować aplikacje do:

- pozyskiwania i analizy obrazu,
- detekcji elementów o określonym polu powierzchni i kształcie (rys. 15.),
- podejmowania decyzji typu *dobry/ zły, odnaleziono defekt/ wolny od defektu* na podstawie posiadanego wzorca (rys. 16.),
- wykrywania elementów o zadanym profilu geometrycznym (rys. 17.),
- wykrywania elementów o niezgodnym schemacie kolorów w zaznaczonym regionie (rys. 18.).

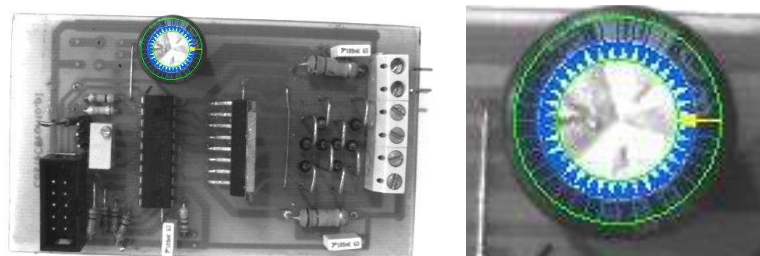
Po skalibrowaniu kamery istnieje także możliwość, pośredniego pomiaru parametrów geometrycznych obiektu przeznaczonego do inspekcji poprzez procedurę wykrycia krawędzi, zliczenia liczby pikseli między krawędziami i przeliczenia uzyskanej wartości na jednostki układu SI .



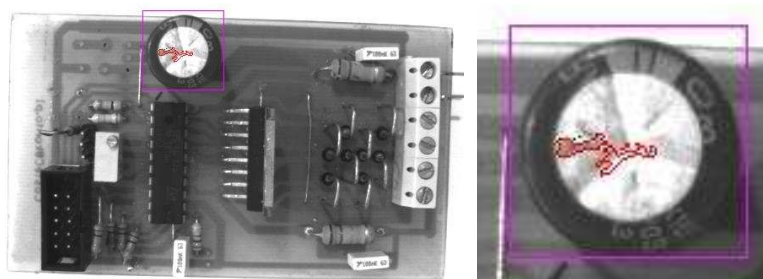
Rys. 15. Wykrywanie przykładowych elementów elektronicznych o określonym polu powierzchni



Rys. 16. Wykrywanie defektów kondensatora na podstawie posiadanego wzorca



Rys. 17. Wykrywanie obiektów o zadanym profilu geometrycznym (okrąg, kwadrat)



Rys. 18. Wyodrębnianie obszarów niezgodnych z wzorcem schematu kolorów

Charakterystyczna cecha VB, polega na uproszczeniu procesu tworzenia nowych aplikacji przeznaczonych dla różnorodnych potrzeb automatycznej inspekcji (identyfikacji) uzależnionych jedynie od zastosowania systemu wizyjnego oraz identyfikowanego obiektu. Zastąpienie żmudnego procesu programowania, przyjaznym dla użytkownika środowiskiem zorientowanym obiektowo, grupującym narzędzia wraz z umożliwieniem ich użycie, bez utraty funkcjonalności wbudowanych algorytmów i narzędzi analizy obrazu umożliwia szerokie zastosowanie w procesach automatycznej identyfikacji systemów widzenia maszynowego opartych o systemy wbudowane zgodne z systemem Windows.

5. PODSUMOWANIE

W artykule przedstawiono architekturę systemów wbudowanych skompilowanych na kodzie dwóch różnych środowisk programowych, dedykowanych do nadzorowania pracy inteligentnej kamery. Funkcjonalność obu środowisk programowych przetestowano na platformie sprzętowej urządzenia inteligentnego. Dodatkowo przeprowadzono kilkanaście testów wbudowanych aplikacji oraz ich możliwości. Na podstawie analizy zgromadzonego materiału i syntezy zdobytej wiedzy, można wnioskować iż funkcjonalność współczesnych urządzeń (zwłaszcza urządzeń inteligentnych) jest determinowana poprzez oprogramowanie i jego możliwości.

O przykładzie funkcjonalności oprogramowania dedykowanego dla urządzeń inteligentnych świadczy sukces pierwszego portalu udostępniającego odpłatnie aplikacje dedykowane dla urządzeń mobilnych firmy Apple. Z początkiem 2011 roku (po dziesięciu latach działalności) wskaźnik liczby pobrań oprogramowania ze strony *AppleStore* [1] przekroczył wartość ponad dziesięciu miliardów, przekraczając tym samym liczbę populacji ludności.

Wielomodułowość oprogramowania przystosowanego dla potrzeb konkretnego użytkownika, oprogramowania zgodnego z indywidualnymi potrzebami klienta, rozszerzającymi możliwości całego urządzenia wraz z jednoczesnym realizowaniem funkcji podstawowych, wyznacza kierunek rozwoju współczesnych systemów wbudowanych nadzorujących pracę urządzeń inteligentnych.

Podziękowanie

Praca badawcza sfinansowana ze środków budżetowych na naukę w latach 2008-2011.

BIBLIOGRAFIA

- [1] AppleStore: <http://store.apple.com/us> (strona aktualna na dzień 03-03-2011)
- [2] B&R IPC: *Windows XP-Embedded Application Guide*. Project number: AT-30-093920, Electronic document, 95 pages, 2005.
- [3] Belbachir A.N. (Ed.): *Smart Cameras*. Springer, 2009. (ISBN 978-1-4419-0952-7)
- [4] Cetinkunt S.: *Mechatronics*. Danvers, 2007. (ISBN 978-0-471-47987-1)
- [5] Ganssle J.: *Embedded Systems World Class Design*. Elsevier, Burlington 2008. (978-0-7506-8625-9)
- [6] Hornberg A. (ed): *Handbook of Machine Vision*. WILEY-VCH Verlag, Darmstadt, 2006. (ISBN 978-3-527-40584-8)
- [7] National Instruments: *Vision Builder for Automated Inspection*. No 372060A-01, September, 2006.
- [8] Pisarczyk P., Jurkiewicz R., Sadowski M.: *Systemy wbudowane – kompendium*. Elektronik, str. 32-35, kwiecień 2003.
- [9] Smoczek J., Szpytko J.: *Narzędzia typu HMI w układach sterowania środków transportu*. Materiały Konferencji Problemy Rozwoju maszyn Roboczych, Dokument elektroniczny, 8 stron, Zakopane 2007.

- [10] Soloman S.: *Sensors Handbook*. McGraw-Hill Professional, New York, 2009 (ISBN 978-0-07-180570-0).
- [11] Sony: *Image Sensing solutions*. Machine Vision Catalogue 2007, 48 pages, 2007.
- [12] Sony: *Intelligent Camera WindowsXP Embedded Developer Guide*. Electronic document 50 pages, Ver. 3.0, Sony Corporation, Japan 2006.
- [13] Sony: *Intelligent Camera XCI V3/XCI SX1* – technical manual. Electronic document, 37 pages. Sony Corporation, Japan 2006.
- [14] Szpytko J., Hyla P.: *Ontologia i semantyka w inżynierii procesów decyzyjnych eksploatowanych środków transportu*. Materiały VI Konferencji LogiTRANS: Logistyka, Systemy Transportowe, Bezpieczeństwo w transporcie, Dokument elektroniczny 12 stron, Szczyrk, kwiecień 2009.
- [15] Uhl T. (red): *Wybrane problemy projektowania mechatronicznego*. Wydawnictwo Katedry Robotyki i Dynamiki Maszyn, Kraków 1999. (ISBN 83-913104-1-8)
- [16] Wang L., Xi J. (ed.): *Smart Devices and Machines for Advanced Manufacturing*. Springer-Verlag, London 2008. (ISBN 978-1-84800-146-6)
- [17] Wawerek Z.: *Machine vision, widzenie maszynowe albo...* *Pomiary Automatyka Robotyka*, Nr 4, str. 6-7, 2008.