

POLITECHNIKA WROCŁAWSKA  
WYDZIAŁ ELEKTRONIKI

---

KIERUNEK: Automatyka i Robotyka (AIR)  
SPECJALNOŚĆ: Systemy informatyczne w automatyce (ASI)

**PRACA DYPLOMOWA  
INŻYNIERSKA**

Głowica wizyjna dla robota społecznego

Vision head for a social robot

AUTOR:  
Robert Budziński

PROWADZĄCY PRACĘ:  
dr inż. Marek Wnuk, I-6

OCENA PRACY:

*Pracę dedykuję Rodzicom.*

*Składam serdeczne podziękowania Panu dr inż. **Markowi Wnukowi** za nieocenioną pomoc, poświęcony czas oraz wielkie zaangażowanie. Dziękuję również mgr inż. **Janowi Kędzierskiemu** za cenne uwagi techniczne i merytoryczne. Podziękowania składam także **Kołu Naukowemu Robotyków „KoNaR”** oraz **Pracownikom Laboratorium Robotów Społecznych projektu LIREC.***

# Spis treści

<b>1</b>	<b>Wstęp</b>	<b>5</b>
1.1	Sformułowanie wymagań stawianych głowicy . . . . .	5
<b>2</b>	<b>Przegląd istniejących konstrukcji</b>	<b>7</b>
2.1	Systemy wizyjne robotów społecznych . . . . .	7
2.1.1	Kismet . . . . .	7
2.1.2	Mertz . . . . .	8
2.1.3	Nexi . . . . .	9
2.2	Kamery z modulem <i>pan-tilt</i> . . . . .	9
2.2.1	Logitech QuickCam Sphere AF . . . . .	9
2.2.2	Planet ICA-HM230 . . . . .	9
2.2.3	Vivotek PZ7151 . . . . .	10
2.2.4	Media-Tech Camtrack MT-4008 . . . . .	11
<b>3</b>	<b>Realizacja</b>	<b>12</b>
3.1	Modyfikacja kamery . . . . .	12
3.1.1	Uzasadnienie modyfikacji . . . . .	12
3.1.2	Opis modyfikacji . . . . .	12
3.2	Regulatory . . . . .	15
3.3	Sterownik . . . . .	17
3.3.1	Realizacja sterownika . . . . .	17
3.3.2	Generowanie przebiegów PWM . . . . .	19
3.3.3	Interfejs . . . . .	21
<b>4</b>	<b>Oprogramowanie</b>	<b>23</b>
4.1	Program sterownika . . . . .	23
4.1.1	Środowisko <i>CodeWarrior</i> , interfejs TBDML . . . . .	23
4.1.2	Projekt programu . . . . .	24
4.1.3	Protokół . . . . .	25
4.1.4	Funkcje programu . . . . .	26
4.2	Moduł detekcji twarzy . . . . .	27
4.2.1	Biblioteka OpenCV . . . . .	27
4.2.2	Filtracja wyników . . . . .	28
4.2.3	Współpraca z programem sterującym . . . . .	28
<b>5</b>	<b>Badania</b>	<b>31</b>
5.1	Badanie charakterystyk regulatorów . . . . .	31
5.2	Badanie powtarzalności pozycjonowania . . . . .	35
5.3	Badanie detekcji twarzy . . . . .	40

<b>SPIS TREŚCI</b>	<b>2</b>
<b>6 Podsumowanie</b>	<b>44</b>
<b>A Schematy</b>	<b>45</b>
<b>B Opis złącz sygnałowych</b>	<b>48</b>
<b>Bibliografia</b>	<b>50</b>

# Spis rysunków

1.1	Robot społeczny Samuel. . . . .	6
2.1	Robot Kismet (fot. Jared C. Benedict). . . . .	7
2.2	Robot Mertz (MIT). . . . .	8
2.3	Robot Nexi (MIT). . . . .	9
2.4	Kamera internetowa Logitech QuickCam Sphere AF. . . . .	10
2.5	Kamera sieciowa Planet ICA-HM230. . . . .	10
2.6	Kamera sieciowa Vivotek PZ7151. . . . .	11
2.7	Kamera internetowa Media-Tech Camtrack MT-4008. . . . .	11
3.1	Płytką z potencjometrem i złączem FFC ZIF dla modułu <i>pan</i> . . . . .	13
3.2	Płytki z potencjometrami dla modułu <i>tilt</i> . . . . .	13
3.3	Półkula obudowy kamery. . . . .	14
3.4	Nóżka kamery. . . . .	14
3.5	Mechaniczne sprzęgnięcie osi obrotu modułu <i>tilt</i> z osią obrotu potencjometru za pomocą wklejonego metalowego pręta. . . . .	14
3.6	Płytki z gniazdem FFC. . . . .	15
3.7	Regulatory pozycji oparte na płytkach z serwomechanizmów modelarskich. . . . .	15
3.8	Opis wyprowadzeń płytki regulatora. . . . .	16
3.9	Wnętrze głowy robota - rozmieszczenie elementów mechanicznych i elektronicznych. . . . .	16
3.10	Struktura i zasoby mikrokontrolera MC9S12A64. . . . .	18
3.11	Płytką rozszerzeniową dla modułu z mikrokontrolerem. . . . .	19
3.12	Sterownik - moduł mikrokontrolera wraz z płytką rozszerzeniową. . . . .	19
3.13	Oscylogram dwóch przebiegów generowanych przy użyciu timera w trybie OC. Zadana wartość: 1,5ms. . . . .	20
3.14	Oscylogram dwóch przebiegów generowanych przy użyciu przerwania cyklicznego i sterowania stanem linii portu. Zadana wartość: 1,5ms. . . . .	20
3.15	Budowa wnętrza układu MAX485. . . . .	21
3.16	Opis wyprowadzeń sterownika. . . . .	21
4.1	Wykonany interfejs TBDML. . . . .	23
4.2	Okno środowiska <i>Code Warrior</i> z uruchomionym narzędziem <i>Processor Expert</i> . . . . .	24
4.3	Ramka protokołu Dynamixel. . . . .	25
4.4	Przykład ramki: zadanie pozycji 520 z prędkością 45 dla urządzenia o numerze identyfikacyjnym 2. . . . .	26
4.5	Schemat blokowy algorytmu filtrowania wykrytych twarzy. . . . .	29
4.6	Zastosowane oznaczenia kolorystyczne. . . . .	30
4.7	Diagram programu sterującego [19]. . . . .	30

5.1	Płytki testowanych regulatorów (od lewej: Hitec HS 6975 HB, Hitec HS 225 MG, Alturn AAS-313 MG). . . . .	31
5.2	Badanie regulatorów - stanowisko badawcze. . . . .	32
5.3	Odpowiedź skokowa regulatora Hitec HS 6975 HB. . . . .	33
5.4	Odpowiedź skokowa regulatora Hitec HS 225 MG. . . . .	33
5.5	Odpowiedź skokowa regulatora AAS-313 MG. . . . .	33
5.6	Odpowiedź skokowa regulatora Hitec HS 6975 HB. . . . .	34
5.7	Odpowiedź skokowa regulatora Hitec HS 225 MG. . . . .	34
5.8	Odpowiedź skokowa regulatora AAS-313 MG. . . . .	34
5.9	Badanie pozycjonowania - scena testowa. . . . .	35
5.10	Badanie pozycjonowania - stanowisko badawcze. . . . .	35
5.11	Kamera oryginalna, sterowanie impulsowe. Zgodny kierunek pozycjonowania przygotowawczego i powrotnego. . . . .	37
5.12	Kamera oryginalna, sterowanie impulsowe. Przeciwny kierunek pozycjonowania przygotowawczego i powrotnego. . . . .	37
5.13	Kamera oryginalna, sterowanie ciągle. Zgodny kierunek pozycjonowania przygotowawczego i powrotnego. . . . .	38
5.14	Kamera oryginalna, sterowanie ciągle. Przeciwny kierunek pozycjonowania przygotowawczego i powrotnego. . . . .	38
5.15	Kamera wyposażona w kodery położenia. Zgodny kierunek pozycjonowania przygotowawczego i powrotnego. . . . .	39
5.16	Kamera wyposażona w kodery położenia. Przeciwny kierunek pozycjonowania przygotowawczego i powrotnego. . . . .	39
5.17	Przykład detekcji wielu twarzy (7). . . . .	41
5.18	Przykład detekcji wielu twarzy (5). . . . .	41
5.19	Przykład filtracji błędnie wykrywanych twarzy. . . . .	42
5.20	Przykład redukcji błędnie wykrywanych twarzy przez dobór parametrów. . . . .	42
5.21	Funkcja wykrywa jedynie twarze w orientacji pionowej. . . . .	43
5.22	Przykład utrzymywania ufności twarzy. . . . .	43
A.1	Płytki PCB sterownika - warstwa górna i dolna. . . . .	45
A.2	Płytki PCB regulatora. . . . .	45
A.3	Płytki połączeniowe dla potencjometrów i taśm FFC. . . . .	45
A.4	Schemat ideowy sterownika. . . . .	46
A.5	Schemat ideowy płytki regulatora. . . . .	47

# Rozdział 1

## Wstęp

Od czasów, kiedy ludzie zaczęli konstruować roboty, stale dążą do możliwości nawiązania z nimi interakcji. Robota społecznego [1] opisano jako *agenta zdolnego do funkcjonowania w różnorodnej grupie ludzi i robotów. Jest on w stanie rozpoznawać członków grupy i wchodzić z nimi w interakcje społeczne, komunikować się i uczyć od nich. Potrafi postrzegać świat i interpretować go wykorzystując nabyte doświadczenia*. Warunkiem postrzegania świata zewnętrznego jest wyposażenie robota w odpowiednie systemy sensoryczne.

Kontakt wzrokowy, obok mowy, mimiki i gestów, jest niezwykle ważnym elementem wykorzystywanym w podejmowaniu interakcji społecznych. W robocie społecznym role wykrywania i śledzenia twarzy pełni odpowiednio zaprojektowany i oprogramowany system wizyjny.

Celem niniejszej pracy dyplomowej jest opracowanie koncepcji systemu wizyjnego dla robota społecznego. Obejmuje ona sformułowanie wymagań, przegląd rozwiązań stosowanych w istniejących robotycznych głowach, kamerach i modułach *pan-tilt* a także opis przyjętego rozwiązania i implementacji programowej.

Wyniki pracy zostały wykorzystane w konstrukcji robota społecznego Samuel<sup>1</sup> (rys. 1.1), realizowanego przez członków Koła Naukowego Robotyków „KoNaR”<sup>2</sup> działającego przy Wydziale Elektroniki, Politechniki Wrocławskiej.

### 1.1 Sformułowanie wymagań stawianych głowicy

System wizyjny robota społecznego będzie pracował w układzie *kamera + komputer + serwomechanizmy*. Kamery będą pełniły jednocześnie rolę atrap oczu osadzonych w głowie robota. Każda z kamer będzie wykonywać obrót w poziomie (*pan*) oraz pionie (*tilt*), naśladując ruch gałek ocznych człowieka. Robot z dwojgiem oczu stanowi intuicyjne uławtwienie, pozwalające na skupienie uwagi obserwatora na robocie, co odgrywa ważną rolę w kontekstach społecznych [25]. Sam wygląd oczu ma niemały wpływ na ogólne postrzeganie robota. Ludzie wnioskują o posiadanych cechach na podstawie wyglądu fizycznego, co można wykorzystać w celu wywołania u obserwatora pożądanego nastawienia i przeświadczenia o posiadaniu przez robota domniemyanych zdolności. Duże oczy sprawiają, że twarz nabiera dziecięcego, niewinnego wyrazu, który jest kojarzony z uległością, pasywnością, podporządkowaniem i brakiem dominacji. [2]. U robota społecznego cechy te można uznać za pożądane. Spełnienie tego założenia umożliwi jednocześnie wykorzystanie kamer

---

<sup>1</sup>Projekt sfinansowany przez Politechnikę Wrocławską, powstał z okazji 100-lecia Uczelni Technicznych we Wrocławiu.

<sup>2</sup><http://www.konar.pwr.wroc.pl>





Rysunek 1.1 Robot społeczny Samuel.

o większych gabarytach, co znacznie poszerzy zakres poszukiwań odpowiedniego modelu i umożliwi wzięcie pod uwagę urządzeń ze zintegrowanym modułem *pan-tilt*. Wybranie kamery z ruchomą głowicą będzie wymagać zbadania jej skuteczności oraz podjęcia decyzji o wprowadzeniu ewentualnych modyfikacji do konstrukcji. Pod uwagę wzięto możliwość zaprojektowania i wykonania niezależnej elektroniki sterującej głowicą. Sterowanie wymaga wybrania i zaimplementowania odpowiedniego protokołu komunikacyjnego. W projekcie należy mieć na uwadze integrację z pozostałymi elementami robota, takimi jak standardy zasilania czy też interfejsy i protokoły komunikacyjne. Nie bez znaczenia pozostaje ergonomia konstrukcji.

W ramach programowej części projektu przygotowany zostanie moduł detekcji twarzy oparty na implementacji detektora Haara wchodzącej w skład darmowych bibliotek OpenCV w wersji 2.0. Jego podstawowym zadaniem będzie filtracja wyników, zaznaczanie wykrytych twarzy na podglądzie obrazu oraz przekazywanie danych o lokalizacji twarzy do nadrzędnego programu sterującego robotem.

# Rozdział 2

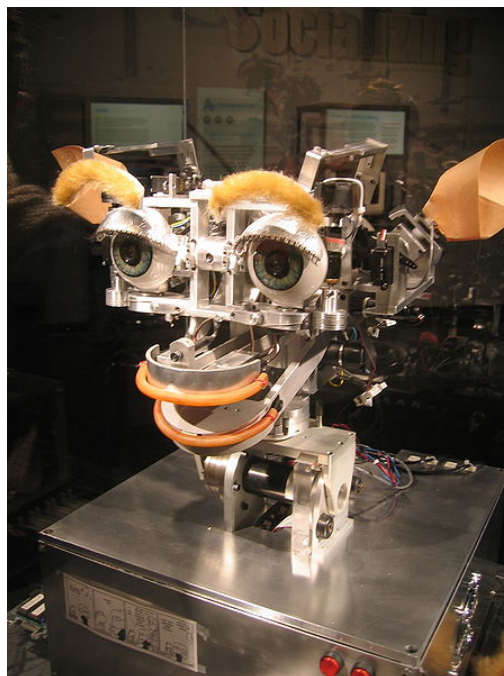
## Przegląd istniejących konstrukcji

### 2.1 Systemy wizyjne robotów społecznych

Znane konstrukcje antropomorficznych robotów społecznych wyposażone są w systemy wizyjne realizujące takie funkcje jak rozpoznawanie twarzy i emocji czy też pomiar odległości. Poniżej przedstawiono trzy roboty społeczne wraz z opisem elementów zastosowanych systemów wizyjnych.

#### 2.1.1 Kismet

Robot Kismet [3] (rys. 2.1) został zbudowany w Massachusetts Institute of Technology - jednej z najbardziej prestiżowych uczelni technicznych świata. Kismet potrafi postrzegać otoczenie wokół siebie - został wyposażony w wizyjny oraz dźwiękowy system sensoryczny. Robot jest w stanie wyrażać szereg podstawowych ludzkich emocji, takich jak gniew, radość, smutek, lęk, zaciekawienie, znużenie, zdziwienie itd.



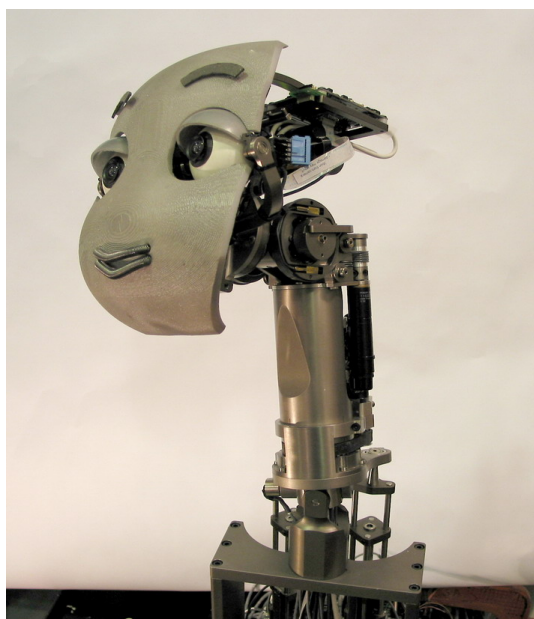
Rysunek 2.1 Robot Kismet (fot. Jared C. Benedict).

System wizyjny Kismet składa się z czterech kolorowych kamer z przetwornikiem CCD. Dwie kamery firmy Elmo Corporation o szerokim polu widzenia (*wide field of view*) zamontowane są centralnie w nosie oraz między oczami. Wyposażone zostały w soczewki o ogniskowej równej 2,2mm. Dzięki obrazowi pobranemu z tych kamer podejmowane są decyzje, na co robot powinien zwrócić uwagę. W oczach zamontowano kamery z przetwornikami CCD, wyposażone w soczewki o ogniskowej równej 8mm. Są to kamery zdolne do pobrania obrazu o wyższej rozdzielczości. Na ich podstawie realizowane jest m.in. wykrywanie twarzy.

Kismet posiada trzy stopnie swobody odpowiadające za ruch oczu oraz trzy stopnie swobody odpowiadające za ruch szyi. Przeguby napędzane są serwomechanizmami z silnikami szwajcarskiej firmy Maxon. Położenie kontrolowane jest przez kodery optyczne o wysokiej rozdzielczości. Robot w sposób wiarygodny naśladuje nawiązanie i utrzymanie kontaktu wzrokowego z obserwatorem.

### 2.1.2 Mertz

Mertz [4] (rys. 2.2), zaprojektowany w Massachusetts Institute of Technology, to robocznica głowa wyposażona w sensoryczny system wizyjny i dźwiękowy. Mertz przeznaczony jest do eksperymentów w kontekście społecznym - umieszczony przez dłuższy czas w miejscu publicznym, usiłuje wejść w interakcje z ludźmi i nauczyć się pewnych zależności.



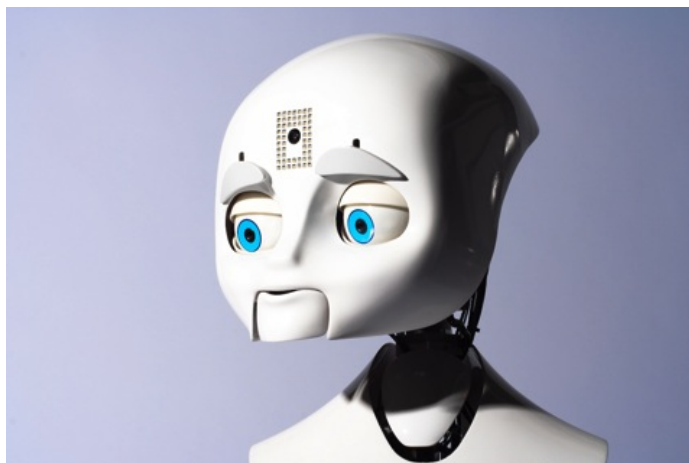
Rysunek 2.2 Robot Mertz (MIT).

Mertz pobiera obraz z dwóch kamer zamontowanych w oczach. Wykorzystane zostały kamery Dragonfly firmy Point Grey wyposażone w przetwornik CCD firmy Sony i interfejs IEEE-1394 (FireWire). Działają one z prędkością 30 klatek na sekundę pobierając obraz o rozdzielczości 640 x 480 i 24-bitowej głębi kolorów. Kamera ta występuje w trzech wersjach różniących się ogniskową obiektywu (4, 6 i 8mm).

Pięć stopni swobody umożliwia ruch gałek ocznych oraz głowy w sposób naśladujący zachowanie człowieka. Ruch zapewniają silniki DC sprzężone z koderami cyfrowymi. Sterowanie prędkością zapewnia zaimplementowany regulator PD.

### 2.1.3 Nexi

Nexi [5] (rys. 2.3) jest autonomicznym robotem mobilnym zbudowanym przez zespół z Massachusetts Institute of Technology. Nexi porusza się dzięki balansującej platformie.



Rysunek 2.3 Robot Nexi (MIT).

Oczy wyposażone zostały w kamery z przetwornikami CCD. W czole umieszczona została podczerwona kamera 3D - CSEM Swiss Ranger (SR-3000) [6]. Jest to kamera typu LRC (*Laser Range Camera*) działająca według zasady TOF (*Time-of-Flight*), dzięki czemu możliwa jest jednoczesna akwizycja obrazu oraz pomiar odległości.

Mechanizm szyi posiada 4 stopnie swobody, dzięki czemu może wykonywać gesty takie jak rozglądanie, kiwanie, drżenie, obserwowanie itp.

## 2.2 Kamery z modułem *pan-tilt*

Na rynku dostępnych jest wiele atrakcyjnych cenowo modeli kamer sieciowych i internetowych wyposażonych w ruchomą głowicę z własnym napędem. Zazwyczaj nie są one w stanie zapewnić dobrej powtarzalności pozycjonowania, ponieważ zostały przewidziane do sterowania przez użytkownika bądź program śledzący. Poniżej przedstawiono cztery modele kamer internetowych i sieciowych wraz z ich podstawowymi parametrami.

### 2.2.1 Logitech QuickCam Sphere AF

Kamera internetowa Logitech QuickCam Sphere AF [7] (rys. 2.4) wyposażona jest w matrycę CCD o rozdzielczości 1,3Mpxl oraz optykę Carl Zeiss z soczewką o ogniskowej równej 3,7mm. Kąt widzenia wynosi 49°. Kamera działa z prędkością 30 klatek na sekundę pobierając obraz o 24-bitowej głębi kolorów. Kamera wyposażona jest w ruchomą głowicę napędzaną silnikami DC zdolną do obrotu w zakresie 140° w poziomie oraz 60° w pionie. QuickCam Sphere AF komunikuje się z komputerem za pomocą interfejsu USB. Cała konstrukcja osadzona jest w plastikowej, kulistej obudowie.

### 2.2.2 Planet ICA-HM230

Kamera sieciowa Planet ICA-HM230 [8] (rys. 2.5) wyposażona jest w matrycę CMOS o rozdzielczości 1,3Mpxl i soczewkę o ogniskowej 4,3mm. Kąt widzenia wynosi 48°. Kamera



Rysunek 2.4 Kamera internetowa Logitech QuickCam Sphere AF.

generuje obraz w standardzie kompresji H.264 przetwarzając 30 klatek w ciągu sekundy. Dzięki wbudowanym silnikom ruch kamery może odbywać się w płaszczyźnie poziomej w promieniu  $355^\circ$  i w pionie w zakresie  $100^\circ$ . Kontrola ruchu odbywa się poprzez sieć TCP/IP. Do komunikacji z komputerem wykorzystywany jest interfejs RJ-45. Kamera obsługuje standard PoE (*Power over Ethernet*), dzięki czemu możliwe jest zasilanie przy wykorzystaniu skrętki.



Rysunek 2.5 Kamera sieciowa Planet ICA-HM230.

### 2.2.3 Vivotek PZ7151

Kamera sieciowa Vivotek PZ7151 [9] (rys. 2.6) wyposażona jest w matrycę CCD firmy Sony oraz zmiennoogniskowy obiektyw o ogniskowej w zakresie 2,8-7,3mm z 2,6-krotnym zoomem optycznym. Kamera przetwarza 30 klatek o rozdzielczości 640 x 480 w ciągu sekundy. Do komunikacji z komputerem wykorzystywany jest interfejs RJ-45. Kamera obsługuje wspomniany wyżej standard PoE. Wyposażona jest w ruchomą, napędzaną głowicę zapewniającą obrót w zakresie  $350^\circ$  w poziomie oraz  $125^\circ$  w pionie.



Rysunek 2.6 Kamera sieciowa Vivotek PZ7151.

#### 2.2.4 Media-Tech Camtrack MT-4008

Kamera internetowa Media-Tech Camtrack MT-4008 (rys. 2.7) wyposażona jest w przetwornik CMOS o rozdzielczości 300Kpxl i plastikowy obiektyw sferyczny. Pracuje z szybkością 30 klatek na sekundę. Z komputerem komunikuje się wykorzystując interfejs USB. Ruchowa głowica wyposażona w silniki krokowe umożliwia ruch w poziomie w zakresie 180° oraz w pionie w zakresie 60°.



Rysunek 2.7 Kamera internetowa Media-Tech Camtrack MT-4008.

# Rozdział 3

## Realizacja

### 3.1 Modyfikacja kamery

W konstrukcji głowicy wizyjnej robota społecznego wykorzystano dwie kamery Logitech QuickCam Sphere AF ze zintegrowanym modulem *pan-tilt*. Kamera ta jest w stanie zapewnić bardzo dobry, ostry obraz. Dzięki sferycznej obudowie nie trzeba dodatkowo stylizować jej na wygląd oka, co było głównym atutem decydującym o wyborze tego modelu.

#### 3.1.1 Uzasadnienie modyfikacji

Zintegrowany moduł *pan-tilt* napędzany jest silnikami prądu stałego. Pozycjonowanie opiera się na zliczaniu czasu ruchu silnika. Jak łatwo się domyślić, zastosowana metoda nie jest w stanie zapewnić wymaganej powtarzalności pozycjonowania, co wykazano podczas badań przedstawionych w rozdziale 5.2. Dodatkowym problemem okazał się niedopracowany sterownik dostarczony przez producenta. Zaobserwowano, że część wysyłanych sygnałów sterujących zwraca komunikat błędu (*Input output error*). Producent nie wyposażył kamery w kodery położenia. Za jedyne sprzężenie zwrotne od pozycji można uznać pobierany obraz. Wykorzystanie go do jednoczesnego pozycjonowania dwóch kamer jest zadaniem niezwykle trudnym i wymagającym znacznej mocy obliczeniowej.

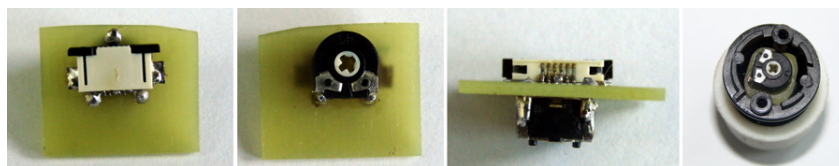
#### 3.1.2 Opis modyfikacji

W celu poprawy parametrów pozycjonowania postanowiono wyposażyć kamery w kodery położenia oraz uniezależnić sterowanie silnikami od sterownika kamery. Poniżej wymieniono dwie rozważane propozycje wraz z ich wadami i zaletami:

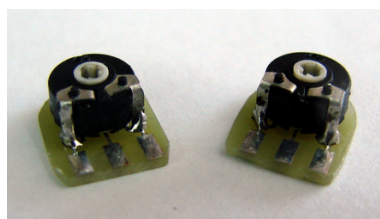
1. Kwadraturowe kodery magnetyczne (np. AS5040 [10])
  - wymagane elementy towarzyszące (dwa kondensatory ceramiczne),
  - bezkontaktowa metoda pomiaru,
  - wymagany magnes o średnicy min. 4mm (zalecana 6mm) namagnesowany diametralnie,
  - konieczność wyprowadzenia 3 przewodów (interfejs PWM) lub 6 przewodów (interfejs inkrementalny, synchroniczny interfejs szeregowy),
  - konieczność zaprojektowania sterownika pozycji dla silników DC.
2. Potencjometr w roli kodera położenia

- brak konieczności stosowania dodatkowych elementów,
- kontaktowa metoda pomiaru, wymagane sprzężenie mechaniczne z osią obrotu,
- konieczność wyprowadzenia 3 przewodów,
- możliwość wykorzystania regulatorów serwomechanizmów modelarskich.

Jako kodery położenia wykorzystano montowane poziomo potencjometry o średnicy 6mm. Głównymi argumentami przemawiającymi za tym rozwiązaniem były mały rozmiar oraz możliwość przystosowania do współpracy z układami regulatorów serwomechanizmów modelarskich. Ze względu na plastikowe tryby przekładni modułu *pan-tilt* konieczne było użycie potencjometru o możliwie najmniejszych oporach ruchu. Odpowiedni model został zdemontowany z jednego z najmniejszych dostępnych serwomechanizmów modelarskich Altum AAS-303 Micro. Potencjometry zostały wlutowane na przygotowane płytki wykonane z laminatu o grubości 1mm, co widać na rysunkach 3.1 i 3.2.



Rysunek 3.1 Płytkę z potencjometrem i złączem FFC ZIF dla modułu *pan*: kolejno widok z dołu, z góry, z boku oraz przycięta płytkę osadzona w nóżce kamery.



Rysunek 3.2 Płytki z potencjometrami dla modułu *tilt*.

Na lewej części półkulistej obudowy kamery, obszar pomiędzy ściankami usztywniającymi uzupełniono klejem Poxipol. Po zastygnięciu wyfrezowano otwór tak, by oś wprowadzonego potencjometru pokrywała się z osią obrotu modułu *tilt*. Płytkę z potencjometrem osadzono w otworze i zabezpieczono przed przesuwaniem się za pomocą „kleju na gorąco”. Przewody ułożono wzdłuż powierzchni wewnętrznej ściany obudowy. Kolejne etapy wprowadzanych modyfikacji obrazuje rysunek 3.3. Potencjometr odczytujący pozycję modułu *pan* zainstalowano w nóżce połączonej z kamerą za pomocą obrotowego przegubu. Płytkę została unieruchomiona za pomocą kleju. Montaż potencjometru wymagał znacznego skrócenia płytki z gniazdem USB oraz zeszlifowania plastikowych elementów pokrywy przykrywającej nóżkę. Zainstalowany koder położenia przedstawia rysunek 3.4.

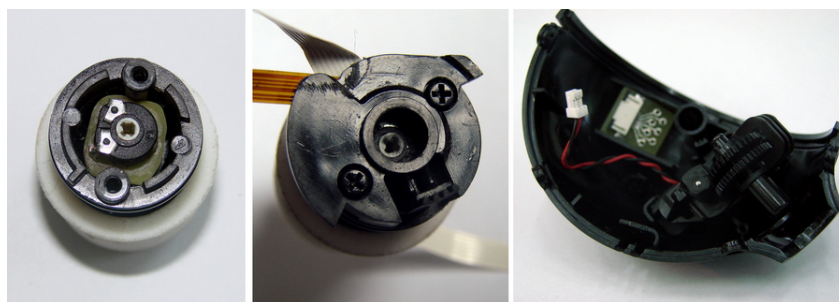
Mechaniczne sprzęgnięcie osi obrotu modułów *pan-tilt* z potencjometrami zapewniają metalowe pręty wklejone w plastikowe elementy. Przykładowe sprzęgnięcie dla modułu *tilt* obrazuje rysunek 3.5. W celu polepszenia skuteczności osadzenia w kleju, pręt został zdeformowany. Luzy na połączeniu wciskowym pręta z potencjometrem zredukowano poprzez wypełnienie otworu montażowego niewielką ilością kleju.



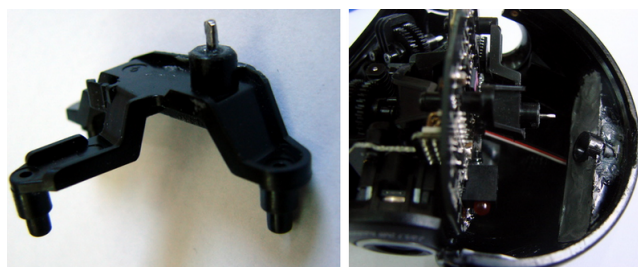
Wyprowadzenia elektryczne silników i potencjometru modułu *tilt* połączone są cienkimi, elastycznymi przewodami z płytką zawierającą złącze FFC ZIF (rys. 3.6). Następnie ośmiożyłową taśmą FFC są poprowadzone do nóżki kamery i wyprowadzone na zewnątrz obok gniazda interfejsu USB. Podobnie osobną sześćżyłową taśmą wyprowadzono linie potencjometru zainstalowanego w nóżce. Taśmy FFC zapewniają odpowiednią elastyczność i trwałość połączenia.



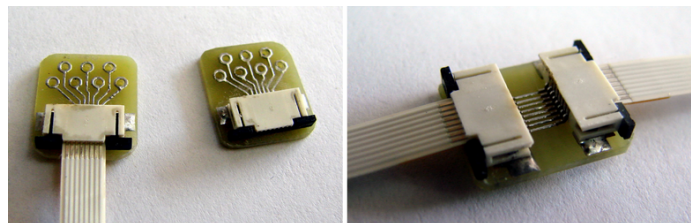
Rysunek 3.3 Półkula obudowy kamery. Kolejno od lewej: oryginalna obudowa; obszar wypełniony klejem oraz przygotowany otwór montażowy; osadzony i zabezpieczony potencjometr wraz z przewodami.



Rysunek 3.4 Nóżka kamery. Kolejno od lewej: osadzenie płytki; nóżka z zamontowaną i zabezpieczoną za pomocą kleju płytką z potencjometrem; mechanizm modułu *pan* sprzęgany z nóżką.



Rysunek 3.5 Mechaniczne sprzęgnięcie osi obrotu modułu *tilt* z osią obrotu potencjometru za pomocą wklejonego metalowego pręta.

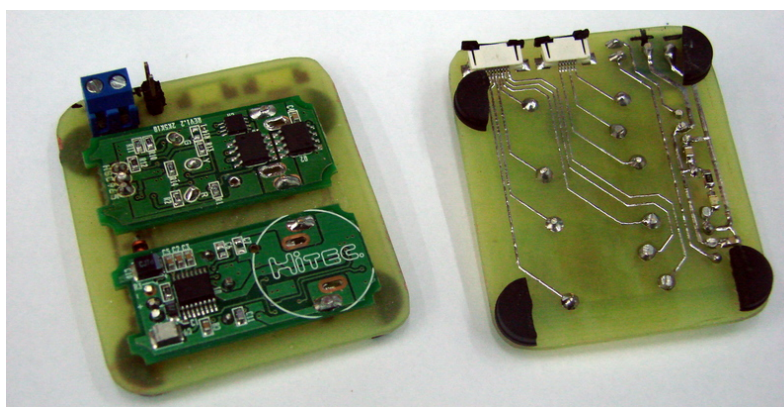


Rysunek 3.6 Płytki z gniazdem FFC. Kolejno od lewej: połączenie taśmy z przewodami potencjometru i silników; płytka łącząca dwie taśmy.

## 3.2 Regulatory

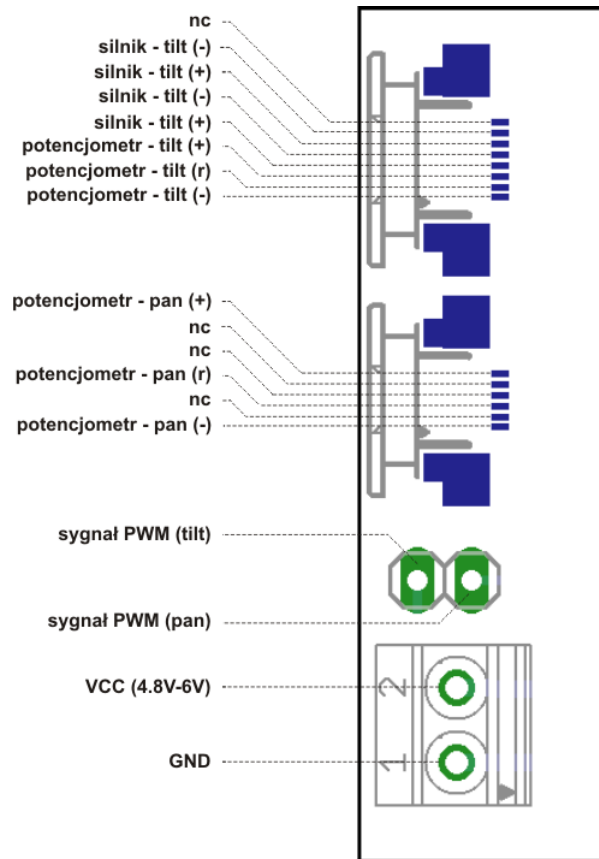
Regulatory (rys. 3.7) pozycji dla modułów *pan-tilt* kamer zostały zrealizowane w oparciu o układy pochodzące z serwomechanizmów modelarskich Hitec HS 6975HB [11], zawierające kompletny cyfrowy regulator PD oraz wyjście mocy oparte na tranzystorach MOSFET. Odpowiedzi impulsowe regulatora współpracującego z modułem *tilt* kamery przedstawiono w rozdziale 5.1.

Pozycja dla regulatora zadawana jest poprzez sygnał PWM (*Pulse-width modulation*), to znaczy sygnał o stałej amplitudzie i zmiennej szerokości impulsu. Przyjęty standard dla regulatorów serwomechanizmów modelarskich definiuje sygnał o częstotliwości 50Hz (okres 20ms) i wypełnieniu w zakresie 1ms-2ms (skrajne pozycje). Regulator odczytuje napięciowy sygnał sprzężenia zwrotnego, uzyskany dzięki zastosowaniu potencjometru w roli regulowanego dzielnika napięciowego. Układ akceptuje napięcie zasilania z przedziału 4.8V-6V.



Rysunek 3.7 Regulatory pozycji oparte na płytkach z serwomechanizmów modelarskich.

Dla opisywanych układów przygotowano płytki PCB, które oprócz organizacji złączy sygnałowych i zasilających, zapewniają filtrację napięcia oraz zabezpieczają układy przed odwrotną polaryzacją zasilania. Regulatory zostały umieszczone w dolnej części głowy robota (rys. 3.9). Połączenie z kamerami zapewniają sześć i ośmiożyłowe taśmy FFC. Rozkład wyprowadzeń płytki regulatora ilustruje rysunek 3.8.



Rysunek 3.8 Opis wyprowadzeń płytki regulatora.



Rysunek 3.9 Wnętrze głowy robota - rozmieszczenie elementów mechanicznych i elektronicznych.

## 3.3 Sterownik

Głównym zadaniem sterownika jest generowanie zadanych przebiegów sterujących dla regulatorów. Sterownik komunikuje się z komputerem wykorzystując wybrany interfejs oraz protokół komunikacyjny. W celu zredukowania liczby połączeń przewodowych oraz elementów elektronicznych wskazana była budowa sterownika, który będzie w stanie obsłużyć zarówno napędy modułów *pan-tilt* oczu jak i serwomechanizmy modelarskie wykorzystane do napędzania powiek oraz ust robota. Napędzające przeguby szyi akтуatory DYNAMIXEL komunikują się w oparciu o strukturę magistralową interfejsu RS-485 [14]. Rozwiązanie to niejako podyktowało użycie RS-485 w sterowniku pozwalając na unifikację sposobu komunikacji robota z komputerem.

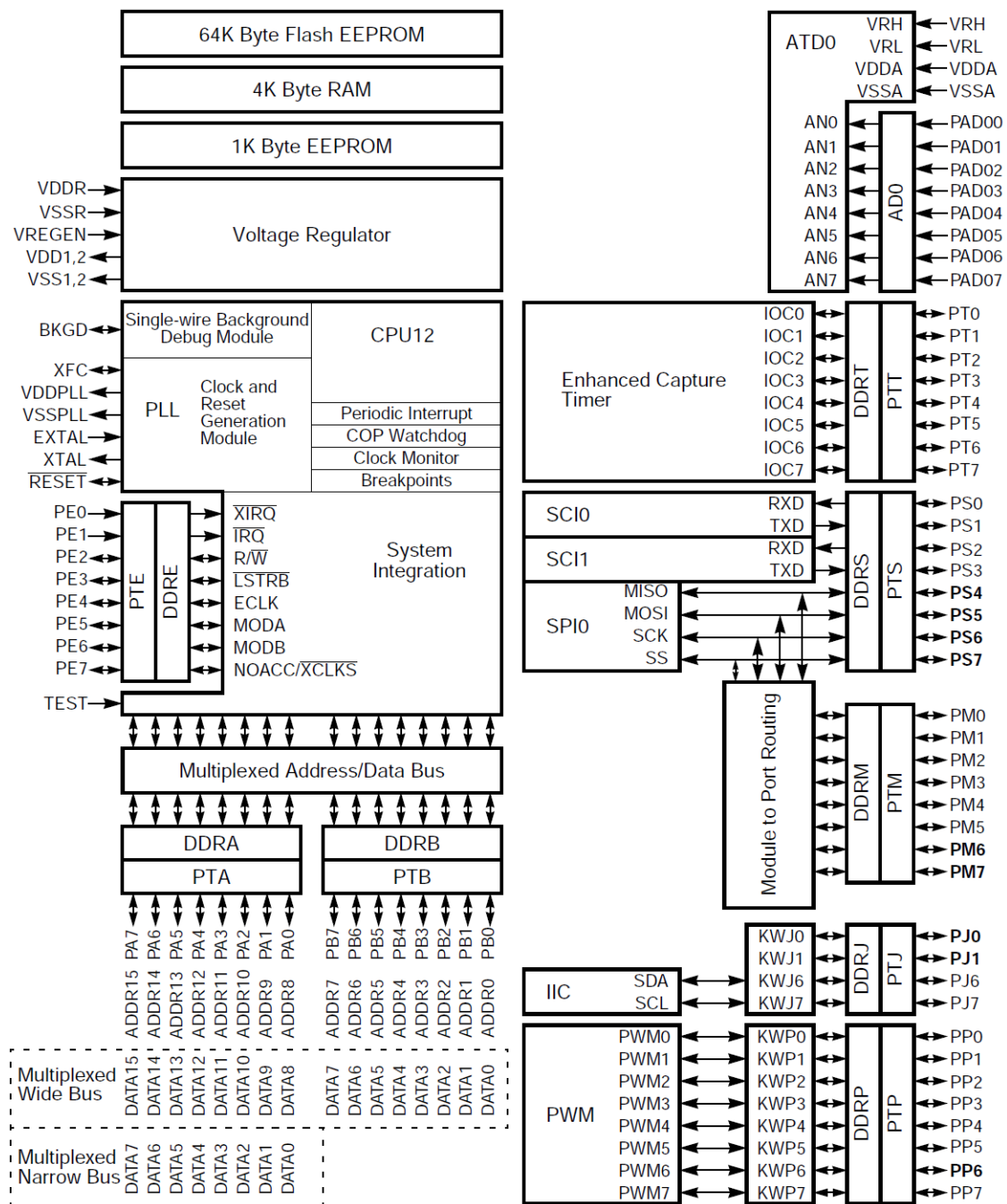
### 3.3.1 Realizacja sterownika

Przy projektowaniu sterownika wzięto pod uwagę poniższe wymagania:

- możliwość generowania 12 przebiegów PWM w rozdzielczości 16 bitów,
- interfejs komunikacyjny RS-485,
- obsługa protokołu DYNAMIXEL,
- uniwersalność,
- osiągalność mikrokontrolera i użytych elementów,
- niski koszt realizacji.

Sterownik został zbudowany w oparciu o moduł z szesnastobitowym mikrokontrolerem MC9S12A64 [15] z rodziny HCS12 firmy Freescale (dawniej Motorola Semiconductors). Jego strukturę i zasoby [16] przedstawia rysunek 3.10. Podstawowe własności zastosowanego modelu mikrokontrolera:

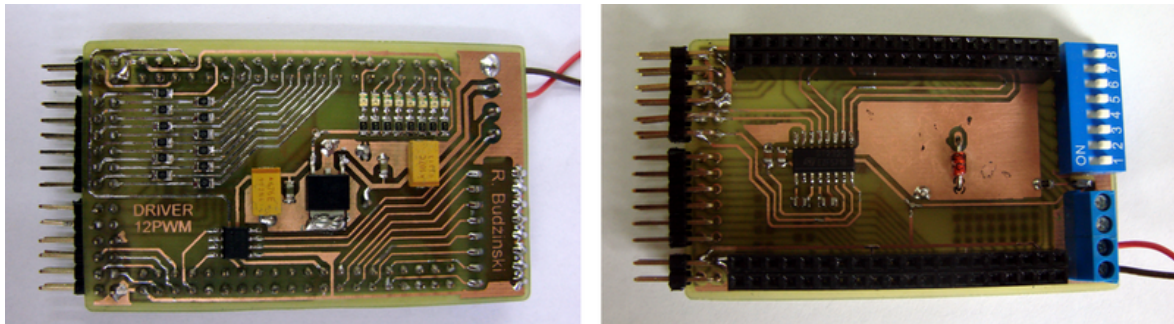
- 16-bitowa jednostka centralna,
- szybkość procesora do 50MHz,
- 64 kilobajty wbudowanej pamięci FLASH, 4 kilobajty wbudowanej pamięci RAM, 1 kilobajt wbudowanej pamięci EEPROM,
- 16-bitowy system timerów (8 kanałów),
- modulator PWM (7 kanałów 8-bitowych lub 4 kanały 16-bitowe),
- dwa asynchroniczne interfejsy szeregowe SCI,
- synchroniczny interfejs szeregowy SPI,
- interfejs zgodny z magistralą I2C,
- napięcie zasilania 3.3V-5.5V,
- interfejs uruchomieniowy BDM (Background Debug Mode),
- obudowa QFP80.



Signals shown in **Bold** are not available on the 80 Pin Package

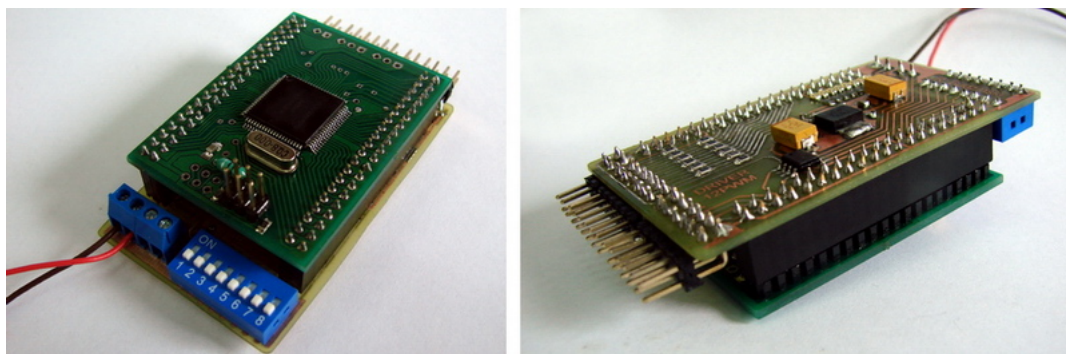
Rysunek 3.10 Struktura i zasoby mikrokontrolera MC9S12A64.

Płytkę modułu zawiera szereg elementów towarzyszących: rezonator kwarcowy, rezystory wymuszające pożądany stan, dławik filtrujący, kondensatory blokujące zasilanie oraz złącze interfejsu uruchomieniowego BDM. Sygnały mikrokontrolera zostały wyprowadzone za pomocą dwóch dwurzędowych listew kołkowych o rastrze 2,54mm.



Rysunek 3.11 Płytkę rozszerzeniową dla modułu z mikrokontrolerem.

W celu zapewnienia oczekiwanej funkcjonalności sterownika przygotowano płytkę rozszerzeniową (rys. 3.11) dla wyżej opisanego modułu. Została ona przystosowana do bezpośredniego połączenia z modułem<sup>1</sup> za pomocą żeńskich gniazd do druku. Płytkę rozszerzeniową zapewnia optymalną organizację wyprowadzeń wykorzystywanych złączy sygnałowych, stabilizację i filtrację napięcia zasilającego dla modułu mikrokontrolera (5V) oraz zabezpieczenie przed odwrotną polaryzacją zasilania. Obecność napięcia zasilającego sygnalizuje zielona dioda LED, zaś aktualny stan sterownika obrazowany jest za pomocą ośmiu diod umiejscowionych na dolnej stronie płytki. Wbudowany 8-sekcyjny przełącznik typu dip-switch umożliwia zmianę numeru identyfikacyjnego ID modułu bez konieczności przeprogramowania mikrokontrolera.

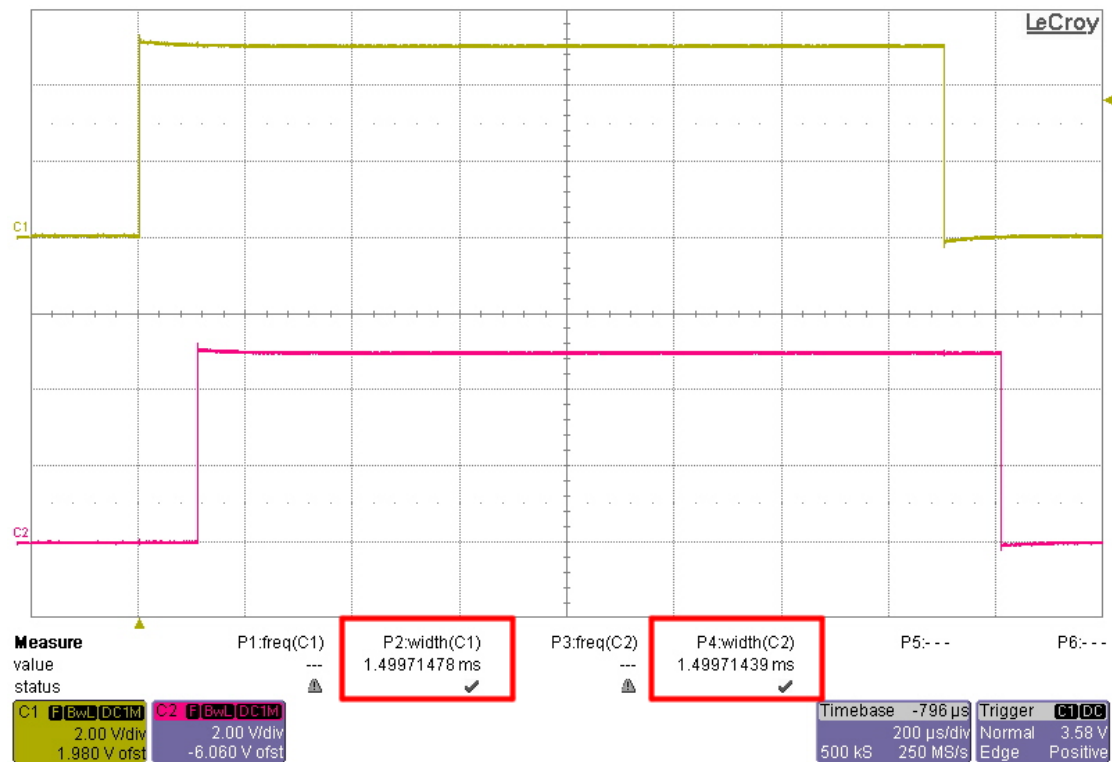


Rysunek 3.12 Sterownik - moduł mikrokontrolera wraz z płytką rozszerzeniową.

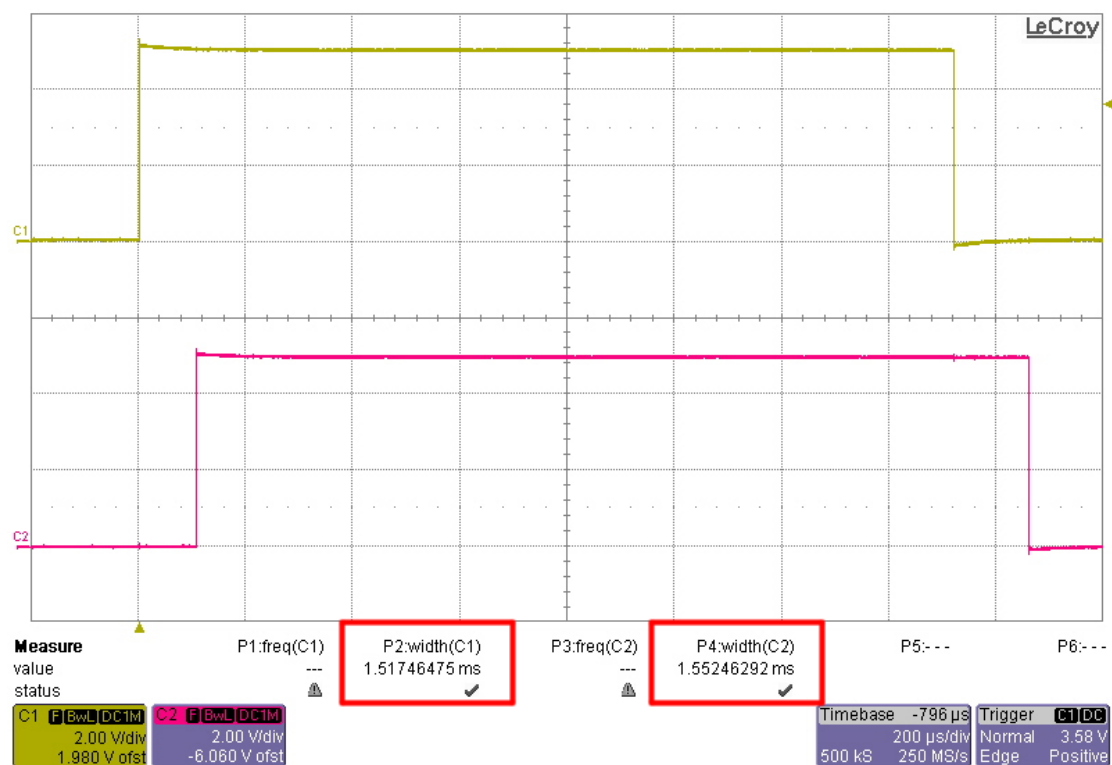
### 3.3.2 Generowanie przebiegów PWM

Sterownik generuje do dwunastu przebiegów PWM o rozdzielczości 16 bitów. Użyty mikrokontroler wyposażony jest w cztery 16-bitowe kanały PWM, uzyskiwane dzięki powiązaniu dwóch kanałów 8-bitowych. Sygnały są wyprowadzone na wyjścia PP1, PP3, PP5,

<sup>1</sup>Wykorzystano bibliotekę programu Eagle autorstwa Edgara Ostrowskiego.



Rysunek 3.13 Oscylogram dwóch przebiegów generowanych przy użyciu timera w trybie OC. Zadana wartość: 1,5ms.



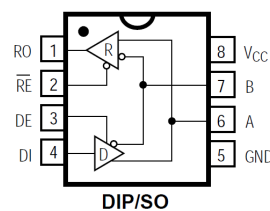
Rysunek 3.14 Oscylogram dwóch przebiegów generowanych przy użyciu przerwanego cyklicznego i sterowania stanem linii portu. Zadana wartość: 1,5ms.

PP7. Pozostałe przebiegi uzyskano dzięki wykorzystaniu timerów. Okres przebiegu generowany jest przez cykliczne przerwanie RTI (*Real Time Interrupt*), które po wywołaniu uruchamia osiem 16-bitowych timerów pracujących w trybie OC (*Output Compare*). Na wyjściach PT0-PT7 pojawiają się impulsy o ustalonym czasie trwania. Zbocze opadające sygnału wywołuje przerwanie powodujące wyłączenie odpowiedniego timera. Dzięki opisanemu rozwiązaniu czas generowanego impulsu odliczony jest możliwie dokładnie, co obrazuje oscylogram przedstawiony na rys. 3.13.

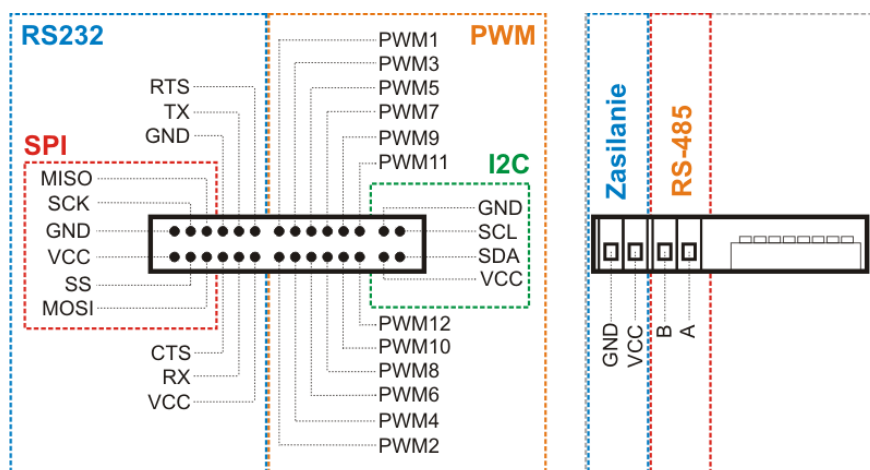
Zastosowane początkowo rozwiązanie polegało na ustawianiu stanu wysokiego na linii portu przy zgłoszeniu przerwania od RTI wyznaczającego okres i stanu niskiego przy przerwaniu od timera wyznaczającego wypełnienie sygnału. Rozwiązanie to nie dało zadowalających rezultatów: czas trwania stanu wysokiego różnił się od zadanego. Ponadto przy ustaleniu jednakowego wypełnienia kolejne przebiegi nieznacznie różniły się między sobą, co widać na rys. 3.14.

### 3.3.3 Interfejs

Sterownik wyposażony jest w wyprowadzenie dwuprzewodowego interfejsu RS-485, popularnego głównie w sieciach przemysłowych. Zastosowany specjalizowany układ MAX485 [13] realizuje konwersję poziomów napięć TTL asynchronicznego interfejsu szeregowego na napięciowy sygnał różnicowy i odwrotnie. Budowę wewnętrzną układu przedstawiono na rysunku 3.15. MAX485 przystosowany jest do pracy w trybie *half-duplex* – kierunek transmisji określany jest przez stan linii DE dla nadajnika i !RE dla odbiornika.



Rysunek 3.15 Budowa wewnątrz układu MAX485.



Rysunek 3.16 Opis wyprowadzeń sterownika.



Sterownik został zaprojektowany z myślą o wykorzystaniu w projektach badawczych i dydaktycznych, toteż oprócz wymaganego interfejsu RS-485 został wyposażony w następujące wyprowadzenia:

- synchroniczny interfejs szeregowy SPI (linie MISO, MOSI, SCK, SS),
- interfejs zgodny z I2C (linie SDA, SCL),
- asynchroniczny interfejs szeregowy z możliwością handshakingu i poziomami napięć zgodnymi z RS-232C (linie RX, TX, CTS, RTS).

Do grupy złączy każdego z wymienionych interfejsów doprowadzono linie +5V i GND, co umożliwi ewentualne zasilanie dołączanych urządzeń. Opis wyprowadzeń przedstawia rysunek 3.16.

# Rozdział 4

## Oprogramowanie

### 4.1 Program sterownika

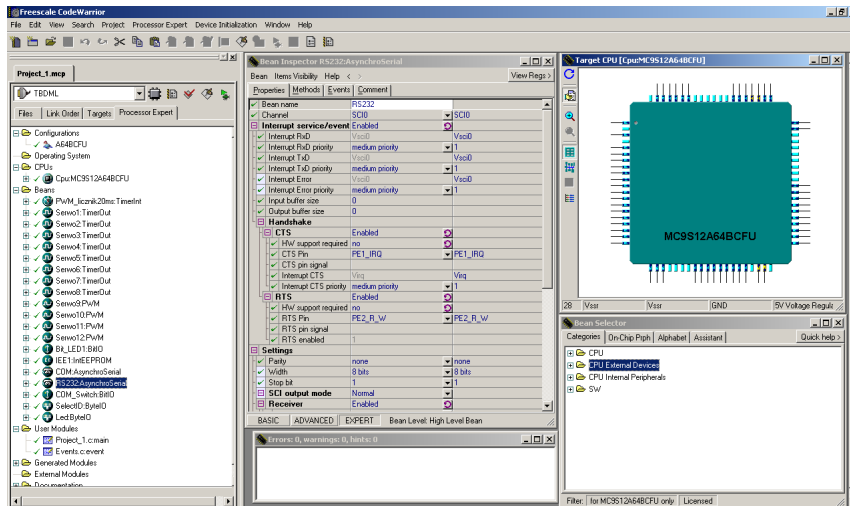
Program sterownika odpowiada za generowanie sygnałów PWM o zadanych wypełnieniach oraz obsługę komunikacji z komputerem za pośrednictwem protokołu Dynamixel. Wykorzystując właściwości regulatora realizuje również sterowanie prędkością ruchu.

#### 4.1.1 Środowisko *CodeWarrior*, interfejs TBDML

Program sterownika został przygotowany w zintegrowanym środowisku *CodeWarrior* [21] dla mikrokontrolerów HCS12(X). Darmowa wersja *Special Edition* umożliwia wygenerowanie do 32kB (język C) lub 1kB (język C++) kodu wynikowego. Dla języka Assembler nie zostały narzucone ograniczenia długości kodu. Środowisko udostępnia szereg narzędzi wspomagających proces tworzenia aplikacji. Narzędzie *Device Initialization* [22] umożliwia wygenerowanie kodu inicjalizującego urządzenia peryferyjne mikrokontrolera oraz obsługę przerwań (np. tablicę wektorów przerwań). *Processor Expert* [22] to narzędzie do szybkiego projektowania aplikacji. Udostępnia szereg zdefiniowanych komponentów odpowiadających za wbudowane urządzenia peryferyjne mikrokontrolera. Pozwala na dostosowanie funkcjonalności każdego elementu w celu jego dopasowania do wymogów aplikacji. Na uwagę zasługuje również debugger *HiWave* [23] pozwalający na analizę programu na działającym docelowym urządzeniu, umożliwiając podglądanie i modyfikowanie rejestrów procesora, zawartości pamięci, zmiennych itp. Na rysunku 4.2 przedstawiono okno środowiska *CodeWarrior* z uruchomionym narzędziem *Processor Expert*.



Rysunek 4.1 Wykonany interfejs TBDML.



Rysunek 4.2 Okno środowiska *CodeWarrior* z uruchomionym narzędziem *Processor Expert*.

Programowanie i debugowanie mikrokontrolera odbywa się poprzez moduł BDM (*Background Debug Mode*). Na potrzeby niniejszej pracy wykonano interfejs TBDML [17] przedstawiony na rysunku 4.1. Urządzenie jest wspierane przez środowisko *CodeWarrior*. Zaletą interfejsu jest możliwość zasilania układu docelowego napięciem 3.3V lub 5V bezpośrednio z magistrali USB, przy czym należy zwrócić uwagę, aby pobór prądu nie przekraczał wartości 100mA. Interfejs może pracować z układem z własnym zasilaniem o napięciu w zakresie 1.2V-5V.

#### 4.1.2 Projekt programu

Program sterownika został przygotowany z użyciem narzędzia *Processor Expert*. Mikrokontroler został skonfigurowany do pracy z zewnętrznym rezonatorem kwarcowym o częstotliwości 8MHz. W celu zwiększenia częstotliwości pracy wewnętrznych urządzeń peryferyjnych mikrokontrolera zastosowano pętlę synchronizacji fazy PLL (*Phase Locked Loop*) uzyskując wartość 25MHz. Zabieg ten pozwolił m.in. na dokładniejsze generowanie przebiegów czasowych w wymaganych przedziałach oraz umożliwił zmniejszenie błędów transmisji dla asynchronicznego interfejsu szeregowego poprzez lepsze dopasowanie częstotliwości mikrokontrolera do prędkości transmisji.

Za pomocą komponentów *PWM* zainicjalizowano cztery kanały sygnałów PWM o rozdzielczości 16 bitów, uzyskanych w wyniku połączenia kanałów 8-bitowych. Dla komponentów *PWM* wygenerowano metody odpowiadające za włączenie/wyłączenie kanałów (*Enable*, *Disable*) oraz zadawanie czasu trwania impulsu (*SetDutyUS*) wyrażonego w mikrosekundach.

Generowanie przebiegów PWM opisane w rozdziale 3.3.2 zrealizowano z wykorzystaniem komponentów *TimerInt* oraz *TimerOut* służących do odmierzenia odcinków czasu. Komponent *TimerInt* odpowiada za odmierzenie okresu (20ms). Źródłem przerwania dla timera jest RTI (*Real Time Interrupt*). Dla komponentu wygenerowano metody odpowiadające za włączenie/wyłączenie licznika (*Enable*, *Disable*) oraz obsługę zdarzenia - wystąpienie przerwania (*OnInterrupt*), po którym następuje uruchomienie ośmiu timerów działających w trybie *OutputCompare* skonfigurowanych za pomocą komponentów *TimerOut*. Timery odpowiadają za wypełnienie sygnału. Dla komponentów wygenerowano

metody odpowiadające za włączenie/wyłączenie timerów (*Enable*, *Disable*), zadawanie odmierzanego czasu (*SetPeriodUS*) wyrażonego w mikrosekundach oraz obsługę zdarzenia - wystąpienie zbocza opadającego (*OnFalling*), po którym następuje wyłączenie timera.

W celu skonfigurowania asynchronicznego interfejsu szeregowego posłużono się komponentem *AsynchroSerial*. Wykorzystano kanał *SC11*, ustawiając następujące parametry transmisji:

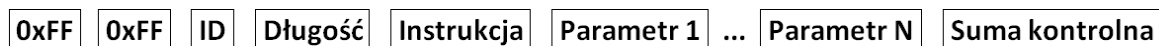
- szybkość transmisji 57600bps,
- 8 bitów danych,
- brak bitu parzystości,
- 1 bit stopu.

Komponent *AsynchroSerial* korzysta z metod odpowiadających za odbieranie/wysyłanie danych (*RecvChar*, *SendChar*, *SendBlock*) a także z obsługi zdarzeń wywoływanych odbiorem danych (*OnRxChar*) oraz zakończeniem wysyłania (*OnTxComplete*). Przystosowanie asynchronicznego interfejsu szeregowego do obsługi standardu RS-485 pracującego w trybie *half-duplex* wymagało użycia dodatkowej linii sterującej kierunkiem transmisji. Za pomocą komponentu *BitIO* zdefiniowano linię 1 portu M jako wyjściową, z domyślnie inicjowanym stanem niskim. Przed wysłaniem danych linia sterująca ustawiana jest w stan wysoki, co powoduje odpowiednie przestawienie trybu pracy układu MAX485. Zakończenie transmisji wywołuje wspomnianą wcześniej obsługę zdarzenia *OnTxComplete* komponentu *AsynchroSerial*, w wyniku czego na linii sterującej pojawia się ponownie stan niski.

Zastosowany w przełączniku 8-sekcyjny przełącznik typu dip-switch umożliwiającą zmianę numeru identyfikacyjnego ID modułu obsłużono za pomocą komponentu *ByteIO* z wygenerowaną metodą *GetVal* odpowiadającą za odczytanie stanu ośmiu linii portu A. Tym samym komponentem zrealizowano obsługę ośmiu diod LED podłączonych do linii portu B. Do sterowania stanem linii wyjściowych wygenerowano odpowiednie metody (*PutVal*, *PutBit*, *SetBit*, *ClrBit*, *NegBit*).

### 4.1.3 Protokół

Wykorzystanie wspólnej magistrali RS-485 do komunikacji ze sterownikiem oraz aktuatorami DYNAMIXEL wymagało zaimplementowania dla sterownika obsługi zgodnego protokołu komunikacyjnego.



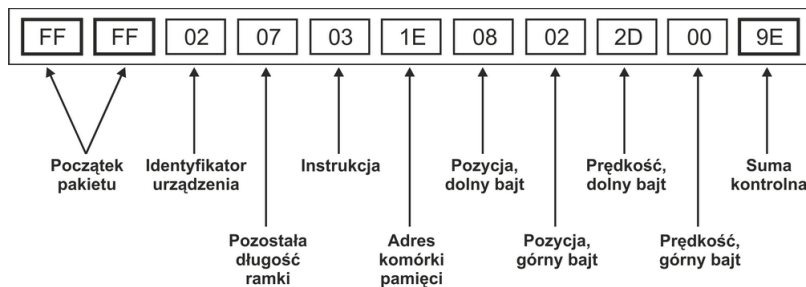
Rysunek 4.3 Ramka protokołu Dynamixel.

Strukturę ramki protokołu DYNAMIXEL [18] przedstawia rysunek 4.3. W pierwszej kolejności wysyłane są dwa bajty o wartości 0xFF informujące o początku pakietu. Trzeci bajt określa identyfikator ID urządzenia, które jest adresatem instrukcji. Identyfikator może przyjmować numer z zakresu 0-253 (0x00-0xFD). Wysłanie ID o numerze 254 (0xFE) spowoduje wykonanie zadanego polecenia przez wszystkie urządzenia. W takim przypadku nie zostanie zwrócony pakiet statusu. Czwartą bajt definiuje pozostałą długość ramki, na którą składają się kolejno: instrukcja (lub kod błędu dla ramki statusu), dodatkowe

parametry (o ile są wymagane) oraz suma kontrolna. Liczba parametrów jest zależna od zadanej instrukcji. Suma kontrolna jest używana do weryfikacji poprawności nadanego pakietu. Obliczana jest wg. następującego wzoru:

$$\text{SumaKontrolna} = \neg(ID + \text{Dlugosc} + \text{Instrukcja} + \text{Parametr1} + \dots + \text{ParametrN})$$

gdzie  $\neg$  oznacza bitowy operator negacji. Jeśli obliczona wartość w nawiasie przekracza 255, pod uwagę należy wziąć osiem najmniej znaczących bitów. Rysunek 4.4 przedstawia ramkę polecenie zadania pozycji o wartości 520 z prędkością 45 dla urządzenia o ID 2.



Rysunek 4.4 Przykład ramki: zadanie pozycji 520 z prędkością 45 dla urządzenia o numerze identyfikacyjnym 2.

Sterownik symuluje 12 urządzeń typu DYNAMIXEL identyfikowanych kolejnymi numerami ID. Numer pierwszego ID ustalany jest za pomocą 8-sekcyjnego przełącznika dip-switch z zastosowaniem notacji binarnej.

#### 4.1.4 Funkcje programu

Sterownik współpracujący z regulatorami bądź kompletnymi serwomechanizmami modelarskimi wykorzystuje ograniczoną ilość instrukcji charakterystycznych dla protokołu DYNAMIXEL. Podczas implementacji obsługi protokołu przewidziano następujące funkcje:

- zadawanie pozycji,
- zadawanie prędkości,
- włączanie/wyłączanie serwomechanizmów,
- ograniczenie zakresu ruchu,
- pingowanie,
- alarmowanie o błędzie sumy kontrolnej,
- włączanie/wyłączanie diody sygnalizacyjnej.

Zadawanie pozycji polega na przełożeniu otrzymanej wartości wyrażonej w postaci liczby z zakresu 0-2000 na wypełnienie sygnału PWM. Funkcja zadawania prędkości wykorzystuje właściwości stosowanych regulatorów PD. Cykliczne podawanie sygnału PWM wskazującego pozycję niewiele dalszą od aktualnej powoduje, że na silniki zostaje podane mniejsze napięcie. Włączanie i wyłączanie serwomechanizmów jest powiązane z generowaniem sygnału PWM. Wyłączenie serwomechanizmu polega na wyłączeniu linii kanału PWM i

pozostawieniu jej w stanie niskim. Sterownik zawiera zaprogramowane skrajne pozycje ruchu. Każdorazowe podłączenie sterownika do zasilania powoduje, że na kanały PWM zostaje wystawiona wartość odpowiadająca pozycji środkowej względem wprowadzonych zakresów. Podczas działania sterownika możliwe jest zadanie nowych wartości pozycji skrajnych. W celach diagnostycznych zaimplementowano również interpretację instrukcji ping. Odebranie instrukcji przez sterownik powoduje wygenerowanie i odesłanie ramki statusu zawierającej informacje dotyczące realizowanego polecenia bądź niemożności jego wykonania.

## 4.2 Moduł detekcji twarzy

Przetwarzanie obrazu wykonywane jest w oparciu o bibliotekę OpenCV [24] w wersji 2.0 pod kontrolą systemu operacyjnego Linux Ubuntu w wersji 9.10. Program został napisany w języku C++. Docelowo został wykorzystany jako moduł w programie obsługi robota. Jego podstawowym zadaniem jest wykrywanie twarzy i przekazywanie jej współrzędnych.

### 4.2.1 Biblioteka OpenCV

OpenCV to zbiór popularnych na całym świecie darmowych bibliotek wykorzystywanych do przetwarzania obrazu. OpenCV został napisany w języku C/C++ dla systemów operacyjnych Linux, Windows oraz Mac OS X. Służy do przetwarzania obrazu w czasie rzeczywistym. OpenCV pozwala na budowanie zaawansowanych systemów wizyjnych wykorzystywanych m.in. w monitoringu i nadzorze, przemyśle, obsłudze interfejsów użytkownika czy też w robotyce. Licencja bibliotek pozwala na realizację komercyjnych produktów.

W celu określenia położenia twarzy posłużono się metodą bazującą na klasyfikatorze Haara. Wykorzystano implementację dostępną w zbiorze bibliotek OpenCV, która korzysta z prostych własności Haara oraz kaskady klasyfikatorów. Wraz z bibliotekami dostarczono gotowe klasyfikatory zapisane w postaci plików XML (*Extensible Markup Language*). W programie wykorzystany został klasyfikator *haarcascade\_frontalface\_alt.xml*. Detekcję twarzy zaimplementowano przy użyciu standardowej funkcji [20]:

```
cvHaarDetectObjects();
```

Funkcję wywołuje się z następującymi parametrami:

```
CvSeq* cvHaarDetectObjects(  
    const CvArr* image,  
    CvHaarClassifierCascade* cascade,  
    CvMemStorage* storage,  
    double scale_factor = 1.1,  
    int min_neighbors = 3,  
    int flags = 0,  
    CvSize min_size = cvSize(0,0)  
);
```

Powyższe parametry definiują kolejno: wejściowy obraz w skali szarości, plik klasyfikatora Haara, przydzieloną pamięć magazynującą, skalowanie, kontrolę zapobiegania fałszywych detekcji, flagę trybu pracy, najmniejszy region, w którym może zostać wykryta twarz. Funkcja zwraca kolejkę zawierającą współrzędne prostokątów tworzących obrysy wykrytych twarzy.

### 4.2.2 Filtracja wyników

Użycie funkcji detekcji twarzy klasyfikatorem Haara wymaga zaimplementowania dodatkowego filtra. Jego zadaniem jest eliminacja zakłóceń polegających na chwilowym zaznaczeniu obiektów błędnie klasyfikowanych jako twarze. Wywołanie funkcji detektora Haara z odpowiednio niskim parametrem skalowania powoduje nasilenie zakłóceń. Schemat blokowy filtra przedstawia rys. 4.5. Wyniki badań detekcji twarzy zamieszczono w rozdziale 5.3.

Zakłócenia, w przeciwieństwie do twarzy, pojawiają się jako krótkotrwałe mignięcia, dzięki czemu można je wyeliminować. Przyjęty algorytm filtracji opiera się na zapamiętywaniu współrzędnych wykrytych twarzy. Podczas każdorazowego wywołania funkcji detekcji następuje sprawdzenie, czy wykryte twarze znajdują się w obszarze twarzy wykrytych wcześniej. Do każdej uprzednio wykrytej twarzy przypisana jest zmienna określająca poziom ufności. Ponowne wykrycie twarzy w kolejnym wywołaniu funkcji powoduje wzrost poziomu ufności oraz zapisanie nowych współrzędnych. Brak twarzy powoduje zmniejszenie zaufania. Twarze, których poziom ufności przekroczył pewien określony próg traktowane są jako wiarygodne. Zaufanie równe 0 powoduje usunięcie twarzy z historii. Zaufanie może rosnąć do określonej wartości. Zastosowanie poziomu ufności zabezpiecza przed chwilowym „zniknięciem” twarzy spowodowanym krótkotrwałym obrotem głowy, zasłonięciem bądź odwróceniem się obserwowanej osoby. Warunkiem utrzymania ciągłości wykrywania twarzy jest nabycie odpowiednio wysokiego poziomu ufności.

Program radzi sobie z wykryciem i identyfikacją wielu twarzy. Maksymalna liczba jest uzależniona od rozdzielczości obrazu oraz przyjętej wielkości obszaru tolerancji. Każda z wykrytych twarzy identyfikowana jest kolejnym, unikalnym numerem. Rozwiązanie takie jest konieczne, gdyż kolejno wywoływany detektor Haara, w zależności od położenia i liczby nadaje twarzom różne numery.

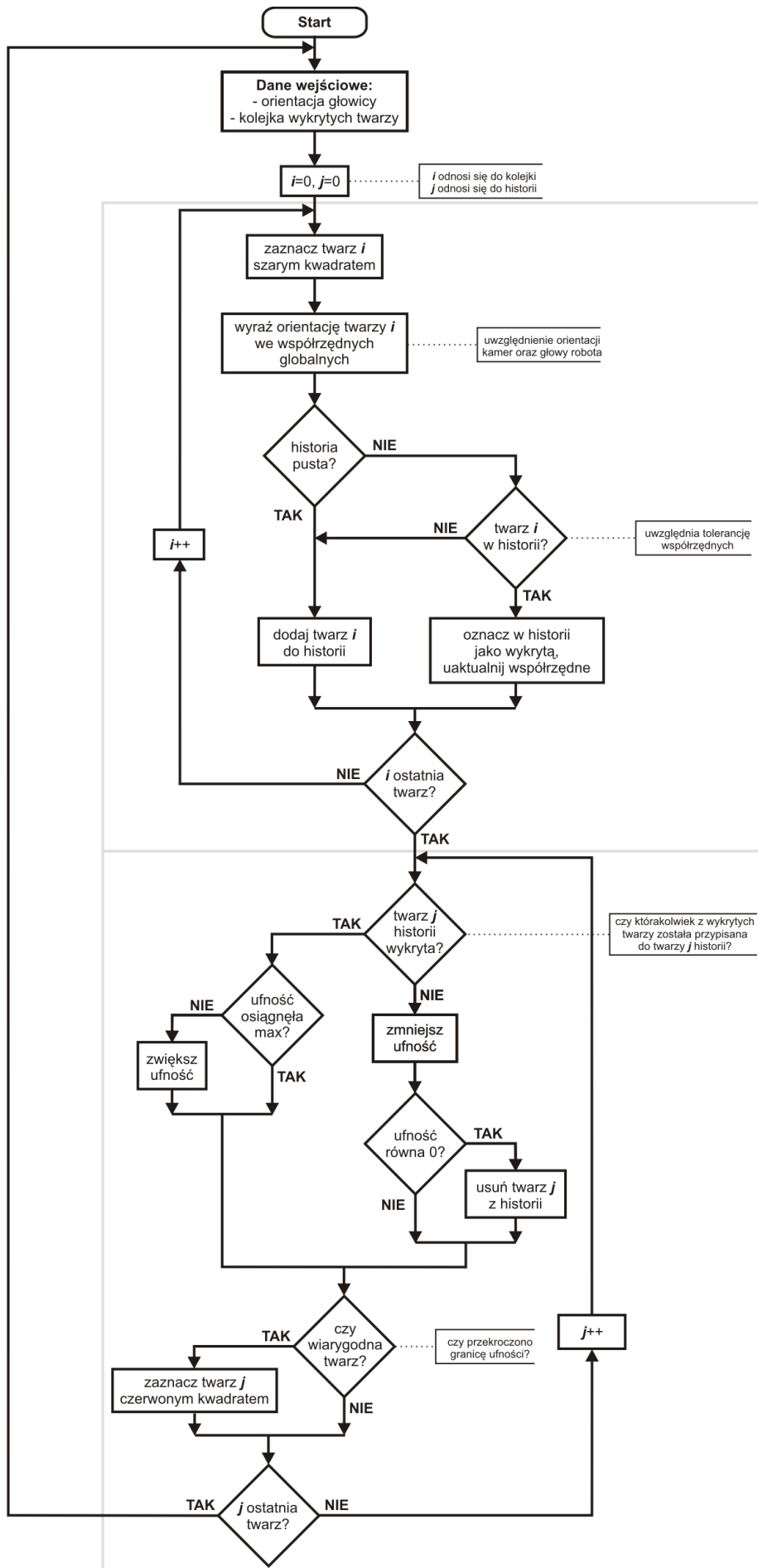
Na pobranym obrazie program rysuje prostokąty (rys. 4.6):

- szary - twarz wykryta przez funkcję *cvHaarDetectObjects*;
- czerwony - twarz wiarygodna;
- żółty - obszar, w którym wiarygodność twarzy zostanie zachowana mimo przemieszczeń.

Dodatkowo program rysuje szare linie dzielące obraz w płaszczyźnie pionowej i poziomej. W miejscu ich przecięcia przyjęto początek układu współrzędnych. Twarze zapisywane są we współrzędnych w układzie globalnym - uwzględniają orientację kamery oraz głowy. Ponieważ osie obrotu kamery i szyi są różne, w celu wyliczenia współrzędnych sferycznych położenia twarzy w układzie globalnym, należy policzyć transformację we współrzędnych jednorodnych. W testowanej implementacji przyjęto uproszczenie polegające na założeniu współosiowości obrotów kamery i głowy. Uznano, że współrzędne z obrazów mają się wprost do przesunięć i obrotów. Za początek układu współrzędnych przyjęto środek obrazu dla centralnego położenia kamery oraz głowy. Na podstawie znajomości kąta widzenia kamery orientacja wykrytej twarzy zostaje wyrażona we współrzędnych sferycznych, a następnie przekazana do programu sterującego robotem.

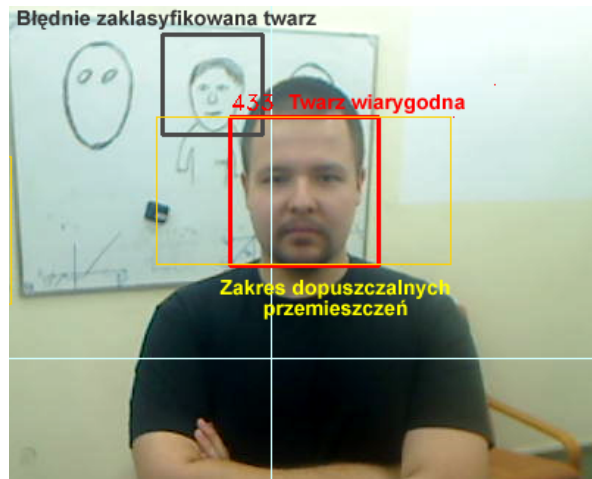
### 4.2.3 Współpraca z programem sterującym

Moduł detekcji twarzy wszedł w skład programu sterującego robotem [19]. Schemat ideowy funkcjonowania programu przedstawia rys. 4.7. Obraz z kamery zostaje przekazany



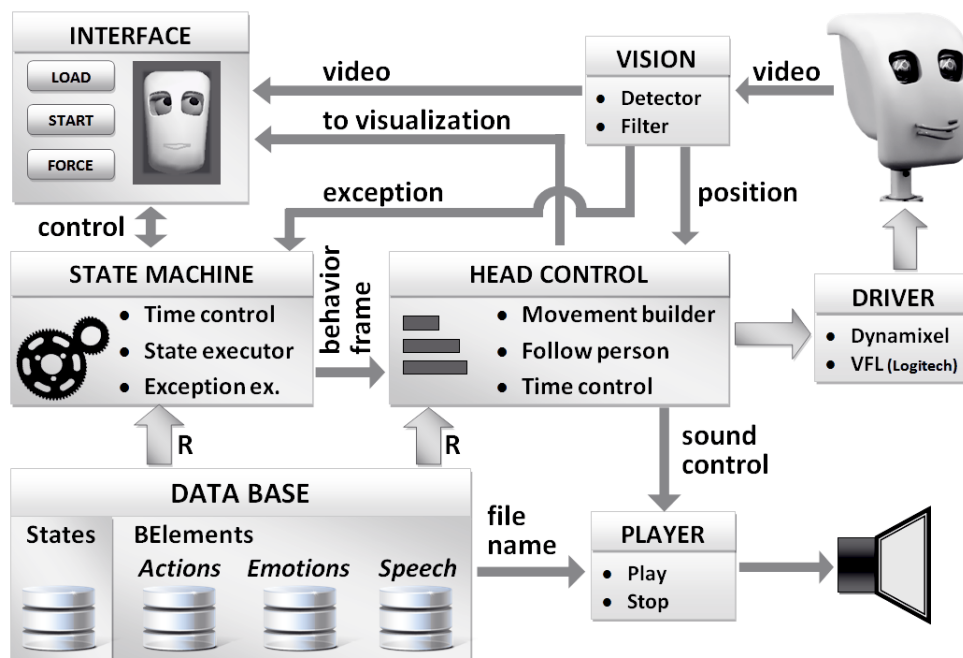
Rysunek 4.5 Schemat blokowy algorytmu filtrowania wykrytych twarzy.





Rysunek 4.6 Zastosowane oznaczenia kolorystyczne.

do modułu *Vision*, gdzie następuje detekcja twarzy i filtracja wyników. Po oznaczeniu wykrytych i zidentyfikowanych twarzy oraz ramek dopuszczalnych przesunięć, obraz zostaje przekazany do modułu *Interface* i wyświetlony na monitorze komputera. Fakt pojawienia się twarzy, która uzyskała wymagany poziom ufności zostaje zgłoszony do automatu stanu *State machine*, natomiast współrzędne twarzy przekazywane są do modułu sterowania głową robota *Head control*, który współpracując z automatem stanu odpowiada za wysterowanie modułów *pan/tilt* kamer oraz aktuatorów DYNAMIXEL. W celu wygenerowania odpowiednich emocji robota, moduł detekcji przekazuje informacje o ilości zauważonych twarzy.



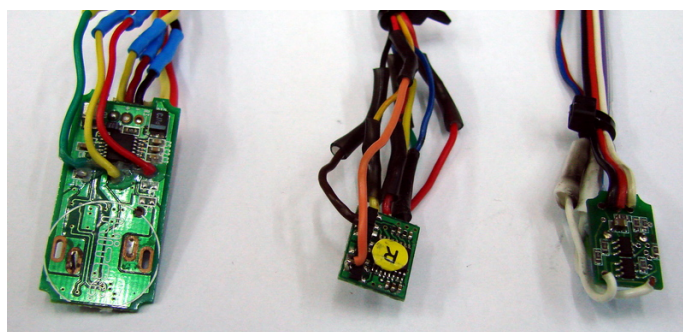
Rysunek 4.7 Diagram programu sterującego [19].

# Rozdział 5

## Badania

### 5.1 Badanie charakterystyk regulatorów

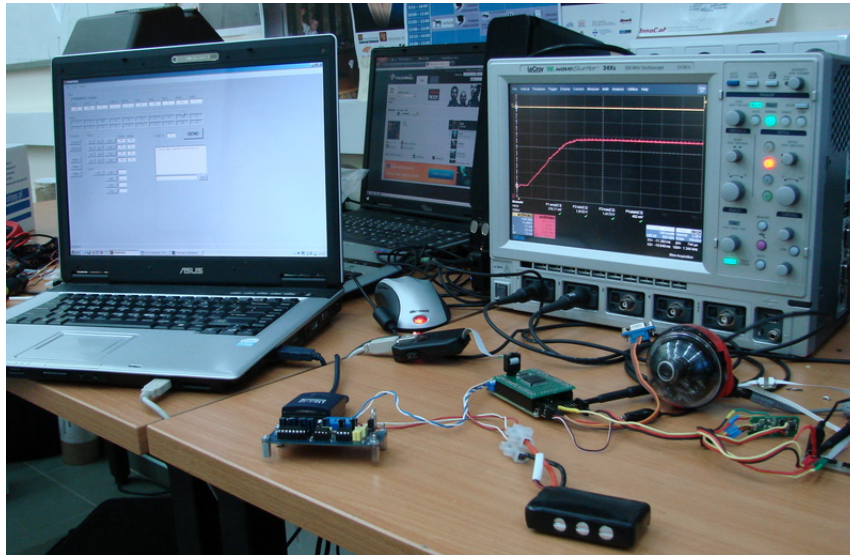
Wybór odpowiedniego regulatora dla kamer został poprzedzony badaniami. Zbadano regulatory pochodzące z serwomechanizmów modelarskich: Hitec HS 6975 HB [11], Hitec HS 225 MG i Altum AAS-313 BB [12]. Płytki regulatorów przedstawia rysunek 5.1. Testy zostały przeprowadzone na module *tilt* kamery z wykorzystaniem oscyloskopu cyfrowego<sup>1</sup> (LeCroy WaveSurfer 24Xs). W celu dokonania analizy regulacji pozycji zarejestrowano odpowiedź skokową regulatorów mierząc napięcie na wyjściu potencjometru sprzężonego mechanicznie z silnikiem elektrycznym napędu *tilt*. Oprogramowanie sterownika zostało przystosowane do wygenerowania sygnału wyzwalającego pomiar w momencie zmiany wypełnienia sygnału PWM. Wartość zadawano za pomocą komputera połączonego ze sterownikiem interfejsem RS-485. Zakres przesunięcia dobrano tak, by różnica napięć na potencjometrze pomiędzy dwoma skrajnymi wychyleniami wynosiła 0,5V (przyjęto tolerancję 1%). Stanowisko badawcze przedstawia rys. 5.2.



Rysunek 5.1 Płytki testowanych regulatorów (od lewej: Hitec HS 6975 HB, Hitec HS 225 MG, Altum AAS-313 MG).

Rysunki 5.3 - 5.8 przedstawiają oscylogramy przebiegów napięć regulatorów. Zielona pionowa linia widoczna na początku przebiegu obrazuje impuls wyzwalający pomiar. Jest on zsynchronizowany ze zmianą wypełnienia sygnału PWM. Zmianę napięcia na wyjściu potencjometru, ilustrującą obrót kamery w pionie, zarejestrowano na przebiegach o kolorze różowym. Oscylogramy regulatorów wykonano z podziałką osi czasu 50ms/div i podziałką osi napięcia 100mV/div.

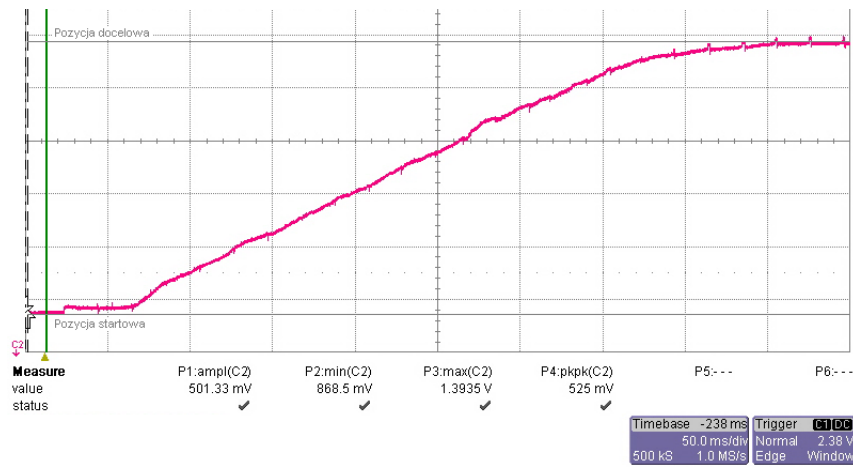
<sup>1</sup>Pomiarów dokonano dzięki uprzejmości pracowników projektu LIREC.



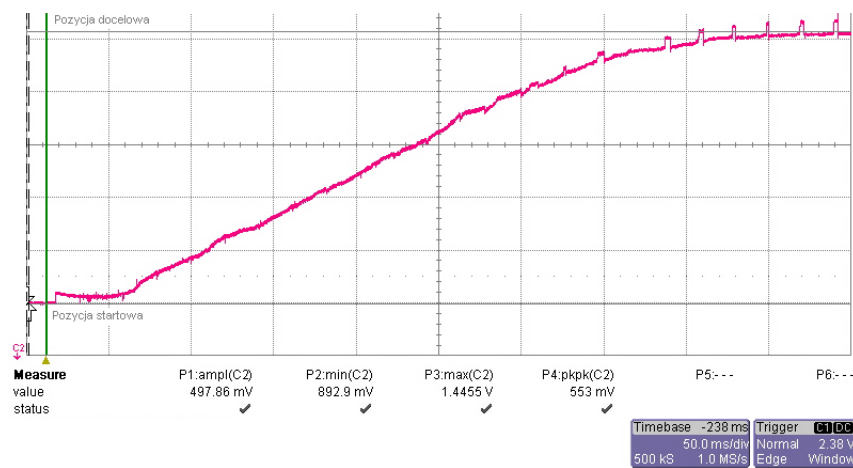
Rysunek 5.2 Badanie regulatorów - stanowisko badawcze.

Na podstawie oscylogramów najlepiej oceniono działanie regulatora serwomechanizmu Hitec HS 6975 HB (rys. 5.3, 5.6). Charakteryzuje się on stosunkowo krótkim czasem regulacji oraz, co najważniejsze, najmniejszymi oscylacjami występującymi w pobliżu pozycji końcowej. Podobnym czasem regulacji charakteryzował się regulator serwomechanizmu Hitec HS 225 MG (rys. 5.4, 5.7), jednakże wykazywał największe oscylacje w końcowej fazie ruchu. Ze względu na najdłuższy czas regulacji oraz widoczne oscylacje, najgorzej oceniono regulator serwomechanizmu Altum AAS-313 MG (rys. 5.5, 5.6).

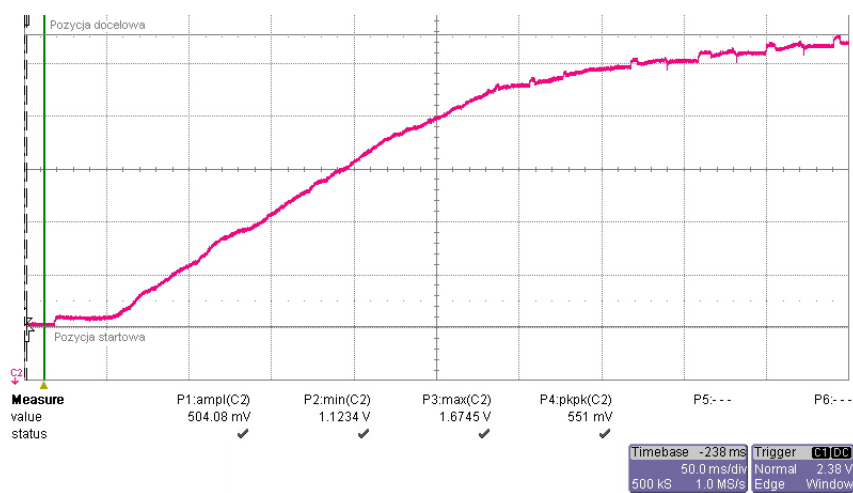
Ostateczną decyzję wyboru regulatora Hitec HS 6975 HB podjęto z uwzględnieniem wizualnej oceny działania poszczególnych regulatorów na module *pan/tilt* kamery.



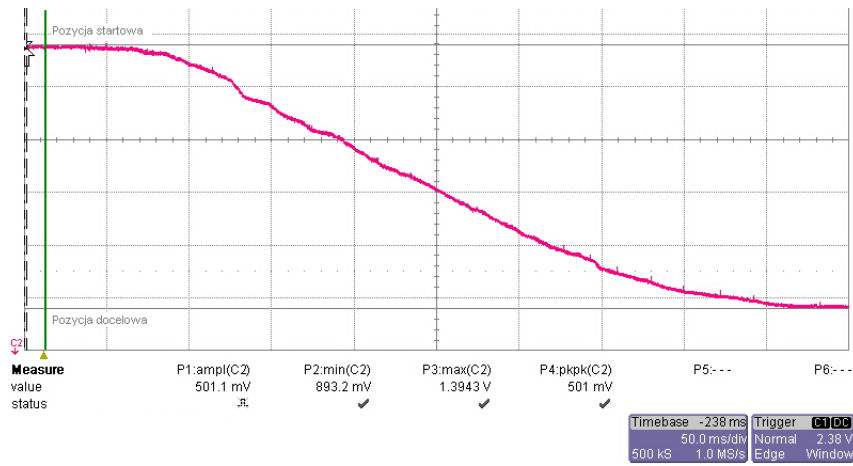
Rysunek 5.3 Odpowiedź skokowa regulatora Hitec HS 6975 HB.



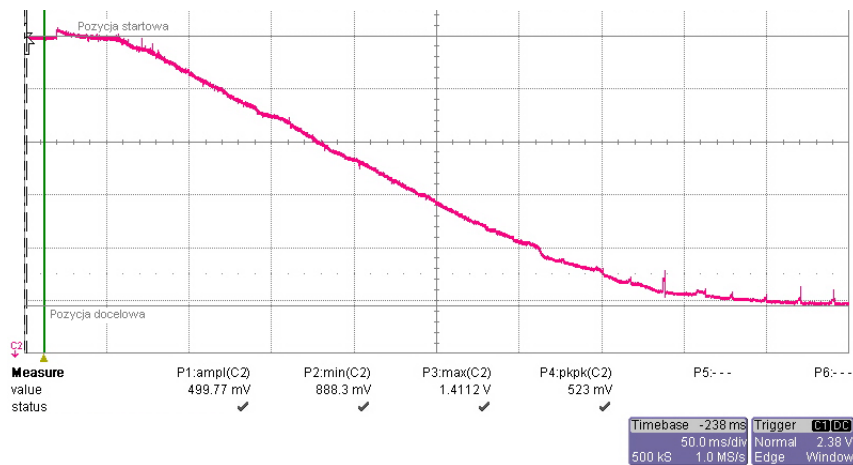
Rysunek 5.4 Odpowiedź skokowa regulatora Hitec HS 225 MG.



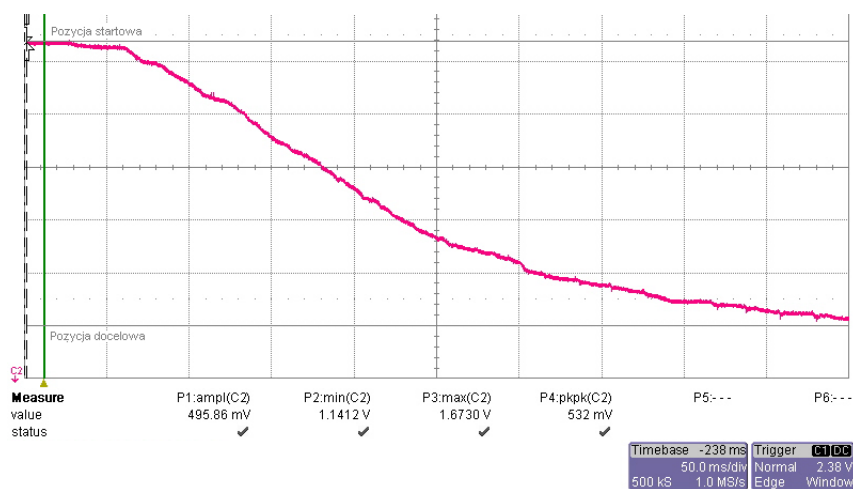
Rysunek 5.5 Odpowiedź skokowa regulatora AAS-313 MG.



Rysunek 5.6 Odpowiedź skokowa regulatora Hitec HS 6975 HB.



Rysunek 5.7 Odpowiedź skokowa regulatora Hitec HS 225 MG.



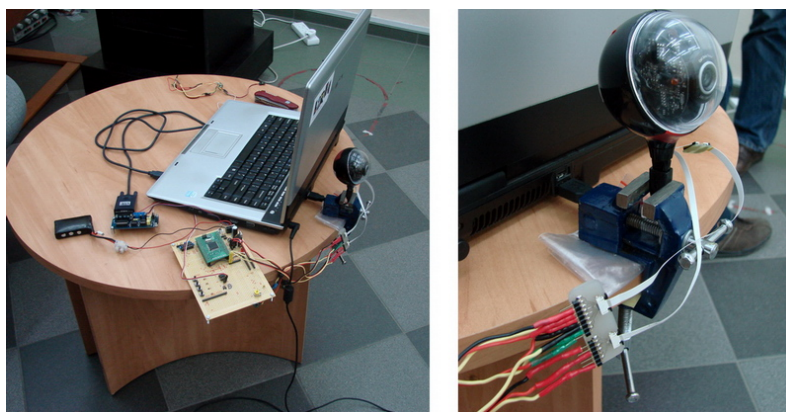
Rysunek 5.8 Odpowiedź skokowa regulatora AAS-313 MG.

## 5.2 Badanie powtarzalności pozycjonowania

W celu zweryfikowania rezultatów wprowadzonych modyfikacji napędów kamer przeprowadzono badanie powtarzalności pozycjonowania modułu *pan-tilt* kamery. Przygotowana została scena testowa (rys. 5.9) z pięcioma czerwonymi znacznikami umiejscowionymi centralnie oraz w narożnikach obrazu. Badanie polegało na porównaniu obrazów pobranych po pozycjonowaniu przygotowawczym oraz po pozycjonowaniu powrotnym, poprzedzonym wychyleniem kamery w okolice pozycji krańcowej. W celu uwidocznienia różnic oba obrazy nałożono z przezroczystością równą 50% zmniejszając ich nasycenie. Na obszarze widocznych na obrazie znaczników nakreślono linie: czerwone, oznaczające pozycjonowanie przygotowawcze oraz niebieskie, oznaczające pozycjonowanie powrotne. Ze względu na luzy występujące na plastikowych mechanizmach modułu *pan-tilt* wprowadzono rozróżnienie na testy wykonane przy zgodnym oraz przeciwnym kierunku dążenia pozycjonowania przygotowawczego i powrotnego. Testy kamery oryginalnej przeprowadzono z wykorzystaniem sterowania ciągłego oraz impulsowego (ciąg impulsów generujących najmniejsze możliwe przesunięcie). Badania powtórzono trzykrotnie. Wyniki zinterpretowano na podstawie najgorszego przypadku.

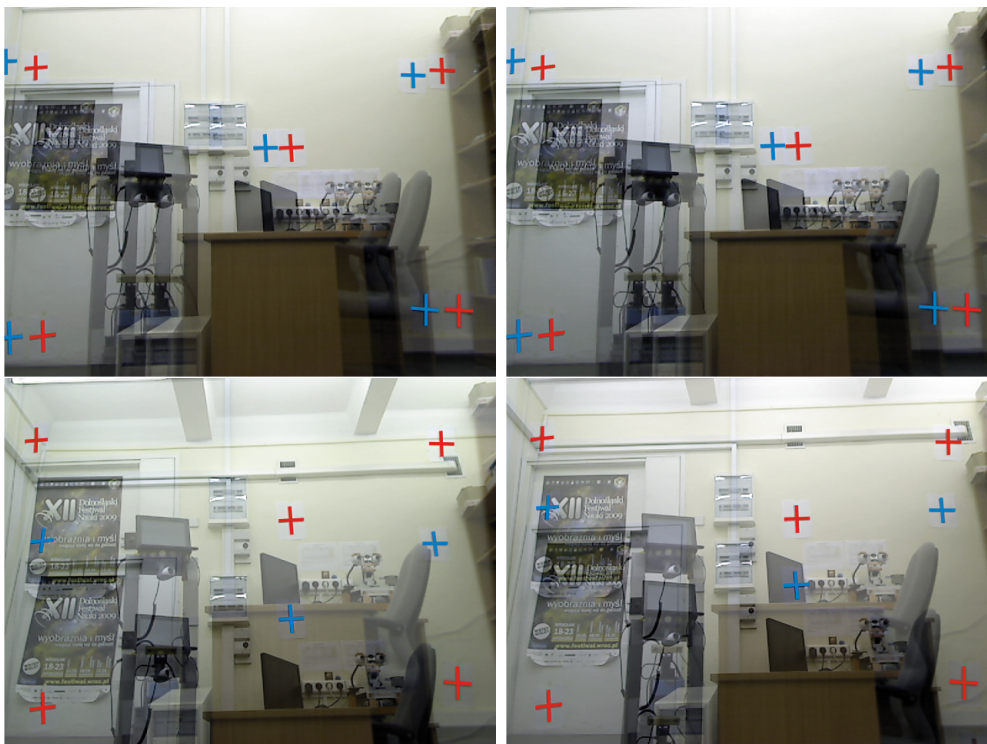


Rysunek 5.9 Badanie pozycjonowania - scena testowa.

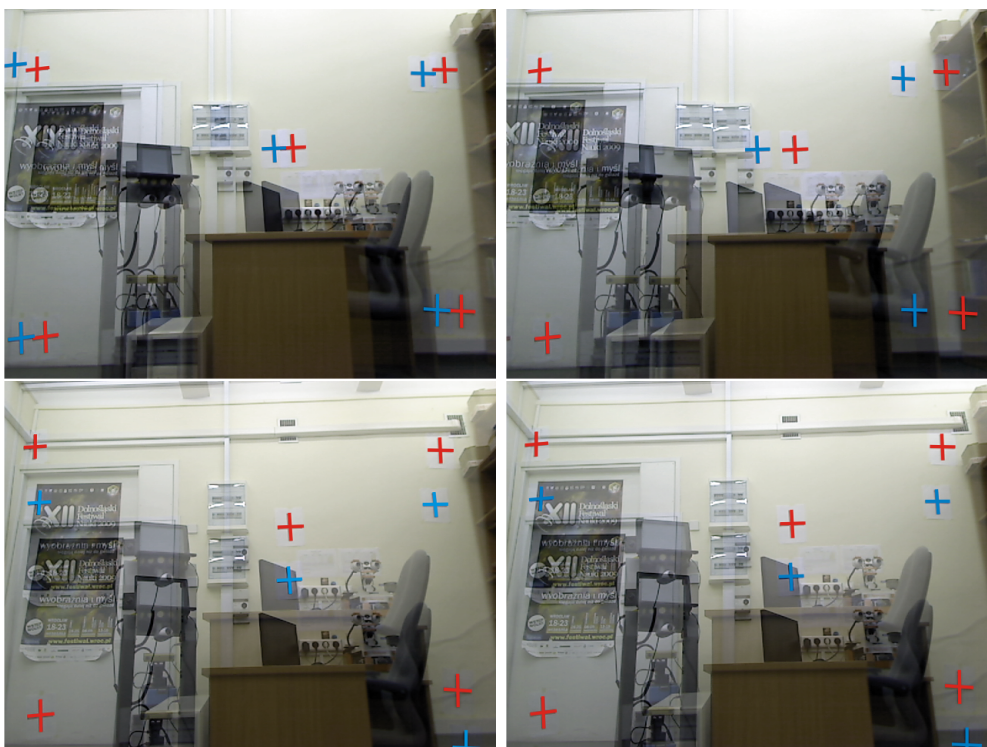


Rysunek 5.10 Badanie pozycjonowania - stanowisko badawcze.

Rezultat testów na kamerze oryginalnej przedstawiają rysunki 5.11, 5.12 (sterowanie impulsowe) oraz 5.13, 5.14 (sterowanie ciągłe). Żadna z przeprowadzonych prób nie dała zadowalających wyników. Niedokładność pozycjonowania była szczególnie rażąca w zakresie ruchu pionowego (tilt). Brak sprzężenia zwrotnego od pozycji powoduje, że niedokładność może wzrastać wraz z ilością wykonanych ruchów. Powtarzalność pozycjonowania zmodyfikowanej kamery wyposażonej w kodery położenia przedstawiają rysunki 5.15, 5.16. W przypadku testu zgodnego kierunku pozycjonowania przygotowawczego i powrotnego, uzyskane obrazy są niemalże identyczne. Niebieski znacznik dokładnie przykrywa znacznik czerwony. Wspomniane wyżej luzy występujące na plastikowych elementach mechanizmów spowodowały, że w teście przeciwnych kierunków pozycjonowania uzyskane obrazy były nieznacznie przesunięte. Przesunięcie to jest całkowicie akceptowalne w opisywanym zastosowaniu.

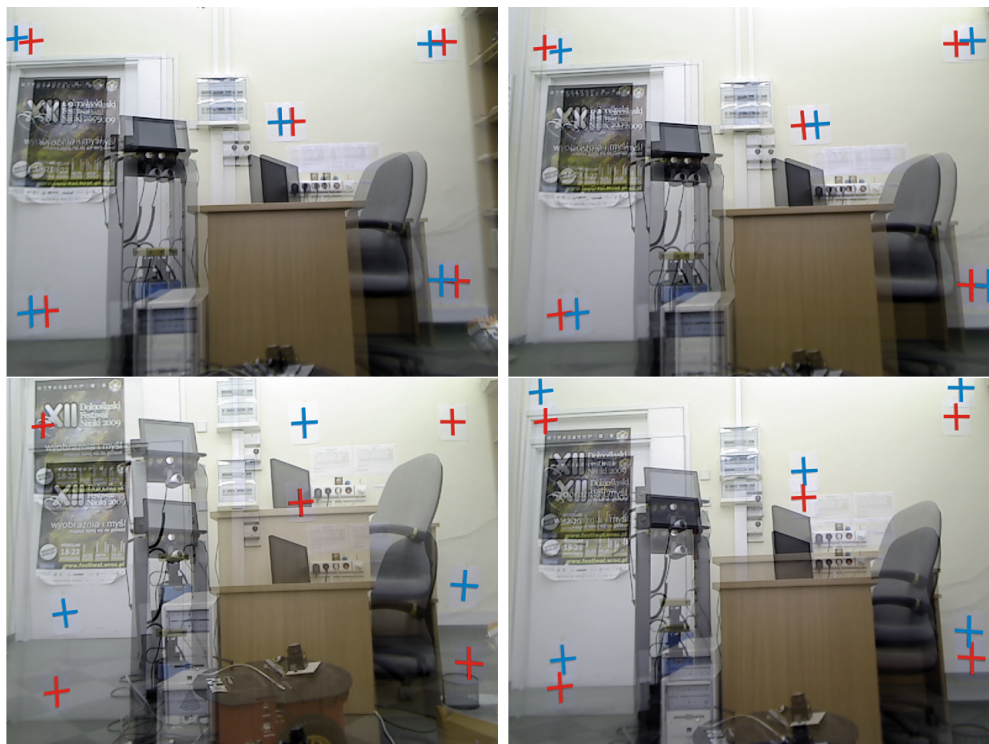


Rysunek 5.11 Kamera oryginalna, sterowanie impulsowe. Zgodny kierunek pozycjonowania przygotowawczego i powrotnego. Pozycjonowanie kolejno od: lewej, prawej, góry, dołu.

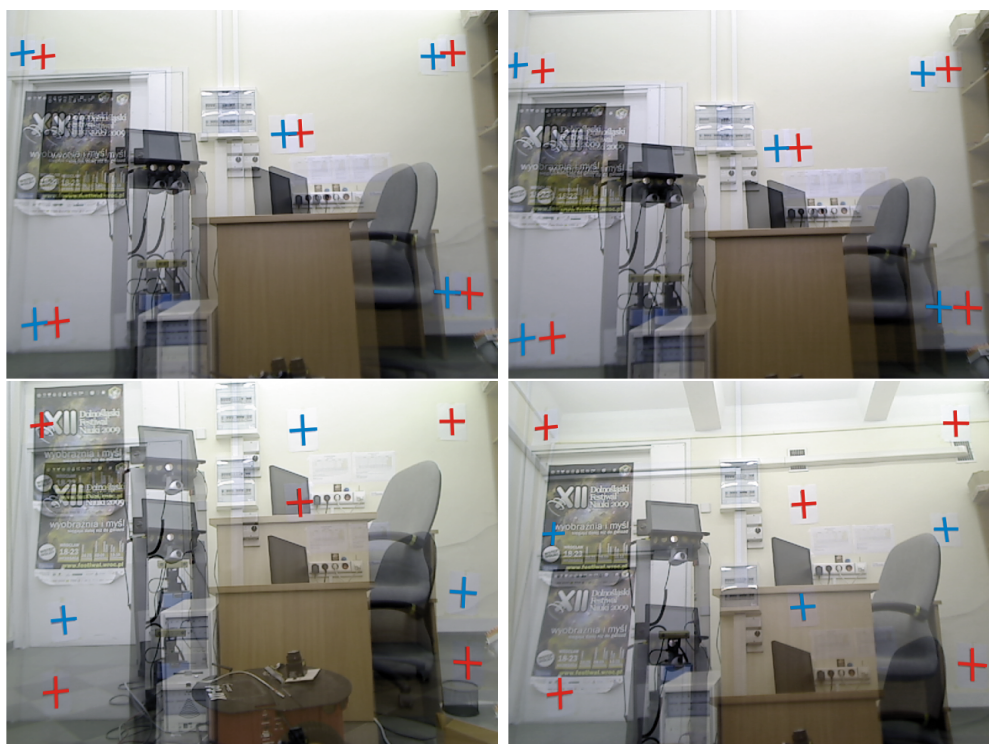


Rysunek 5.12 Kamera oryginalna, sterowanie impulsowe. Przeciwny kierunek pozycjonowania przygotowawczego i powrotnego. Pozycjonowanie kolejno od: lewej, prawej, góry, dołu.





Rysunek 5.13 Kamera oryginalna, sterowanie ciągle. Zgodny kierunek pozycjonowania przygotowawczego i powrotnego. Pozycjonowanie kolejno od: lewej, prawej, góry, dołu.



Rysunek 5.14 Kamera oryginalna, sterowanie ciągle. Przeciwny kierunek pozycjonowania przygotowawczego i powrotnego. Pozycjonowanie kolejno od: lewej, prawej, góry, dołu.



Rysunek 5.15 Kamera wyposażona w kodery położenia. Zgodny kierunek pozycjonowania przygotowawczego i powrotnego. Pozycjonowanie kolejno od: lewej, prawej, góry, dołu.



Rysunek 5.16 Kamera wyposażona w kodery położenia. Przeciwny kierunek pozycjonowania przygotowawczego i powrotnego. Pozycjonowanie kolejno od: lewej, prawej, góry, dołu.

### 5.3 Badanie detekcji twarzy

Poprawność działania detektora twarzy oraz filtru sprawdzono na zaaranżowanej scenie<sup>2</sup>. Program przetestowano z udziałem ludzi, a także z wykorzystaniem fotografii twarzy oraz rysunków przypominających kształtem ludzkie twarze.

W pierwszej kolejności sprawdzono program pod względem zdolności wykrywania wielu twarzy oraz identyfikowania ich kolejnymi, unikalnymi numerami, których wartość wskazuje kolejność pojawiania się na scenie. Najwyższym numerem oznaczana jest twarz, która na scenie pojawiła się jako ostatnia i uzyskała wymagany poziom ufności. Jak widać na rysunkach 5.17 i 5.18 program bezbłędnie wykrył większą liczbę twarzy (kolejno 7 i 5). Również kolejność dołączania twarzy do sceny testowej znajduje odzwierciedlenie w nadawanych numerach identyfikacyjnych.

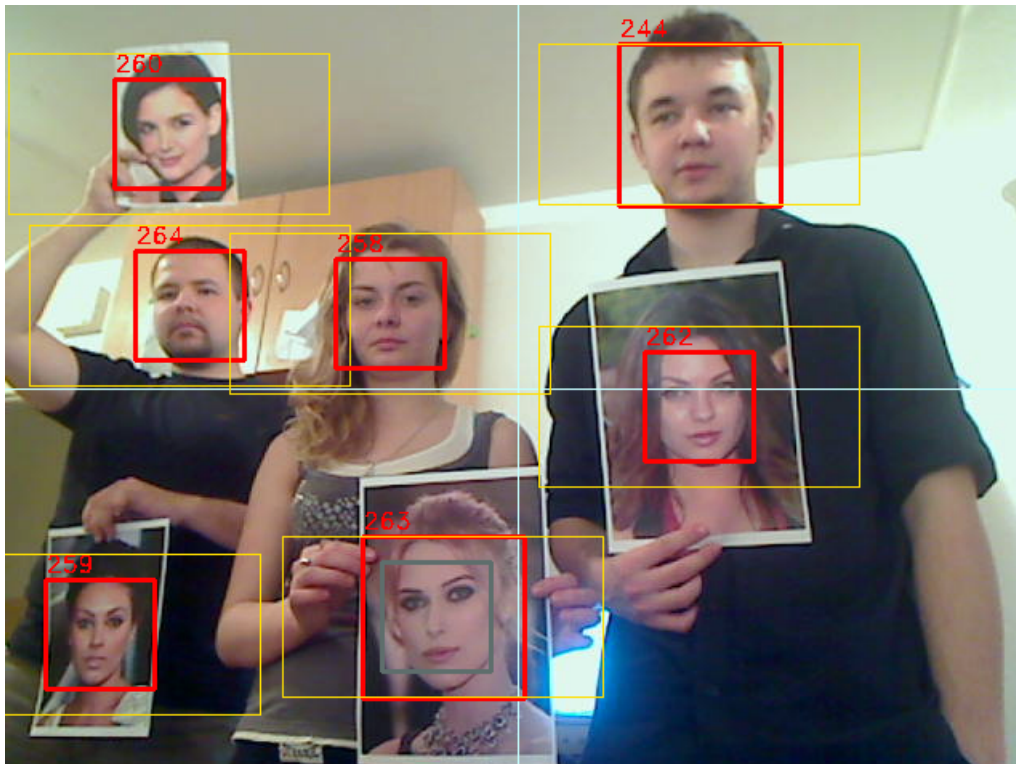
Kolejny test pozwolił na stwierdzenie skuteczności działania filtru fałszywych wykryć twarzy. Jako obiekty wywołujące zakłócenia wykorzystano trzy proste schematyczne rysunki twarzy o różnych liczbach detali. Parametry funkcji *cvHaarDetectObjects* mające największy wpływ na czułość detekcji twarzy, to jest skalowanie oraz najmniejszy region, w którym twarz może zostać wykryta, ustawiono tak, by zakłócenia występowały stosunkowo często: *scalefactor=1.1*, *min\_size=cvSize(20,20)*. Zaimplementowany algorytm prawidłowo odfiltrował występujące zakłócenia od realnych twarzy, co ilustruje rysunek 5.19. Sporadycznie zdarza się, że przy wysokiej czułości obiekt przypominający twarz uzyska wymagany poziom ufności. Opisaną sytuację można wyeliminować dobierając odpowiednio parametry funkcji. Rysunek 5.20 przedstawia detekcję twarzy z parametrami: *scalefactor=3.1*, *min\_size=cvSize(50,50)*. Widoczne na tablicy schematyczne rysunki twarzy nie powodują zakłóceń. Ograniczenie czułości spowodowało, że mniejsza i mniej wyraźna twarz umiejscowiona w dolnym obszarze sceny nie została wykryta.

W programie wykorzystany został klasyfikator *haarcascade\_frontalface\_alt.xml*, w którym zawierają się cechy frontального widoku twarzy. Oznacza to, że program jest w stanie wykryć jedynie twarze usytuowane pionowo frontem do kamery. Tolerowane jest jedynie niewielkie przechylenie. Opisaną właściwość przedstawia rys. 5.21.

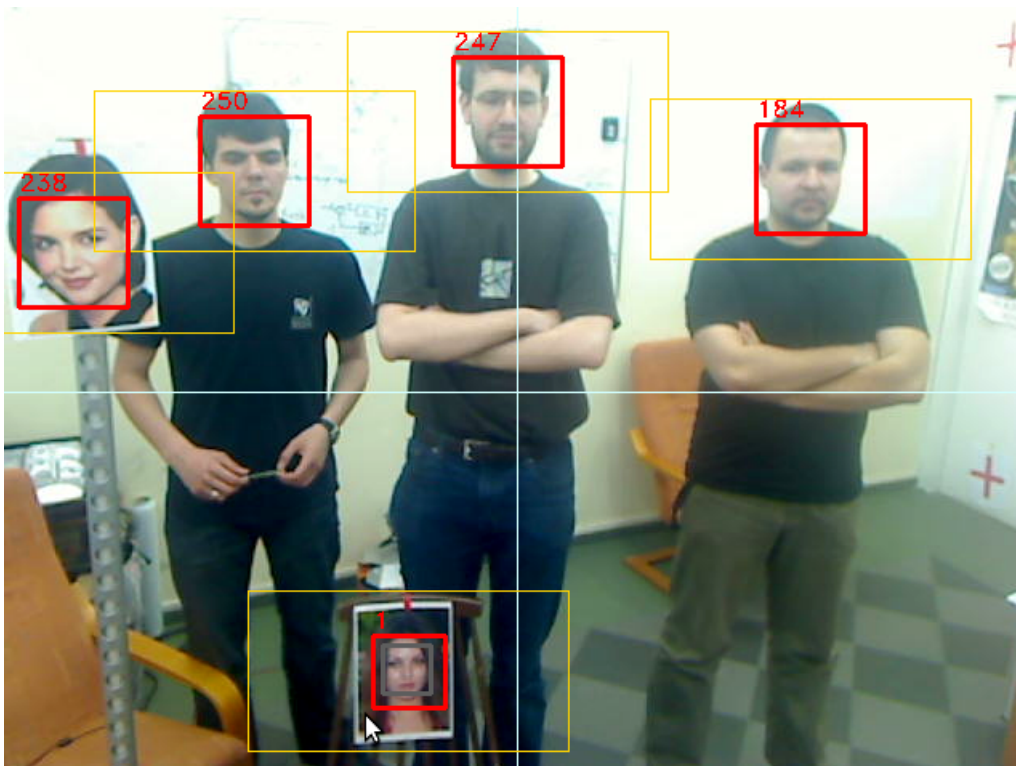
Rzadko zdarza się, by człowiek obserwujący robota stale był zwrócony twarzą w kierunku kamery. Naturalnym zachowaniem jest chwilowe odwracanie się, przekrzywianie głowy, przysłanianie jej dłonią itp. Uzyskanie przez daną twarz wysokiego poziomu ufności powoduje, że jej brak zostanie zaakceptowany dopiero po spadku poziomu ufności do określonego poziomu. W przeciwnym przypadku chwilowe przesłonięcie twarzy spowodowałoby jej ponowne wykrycie, a co za tym idzie nadanie jej nowego numeru identyfikacyjnego i przekazanie do automatu stanu jako nową osobę. Jak pokazuje rys. 5.22, dopiero trwające dłuższą chwilę odwrócenie głowy zostanie uznane za jej faktyczny brak.

---

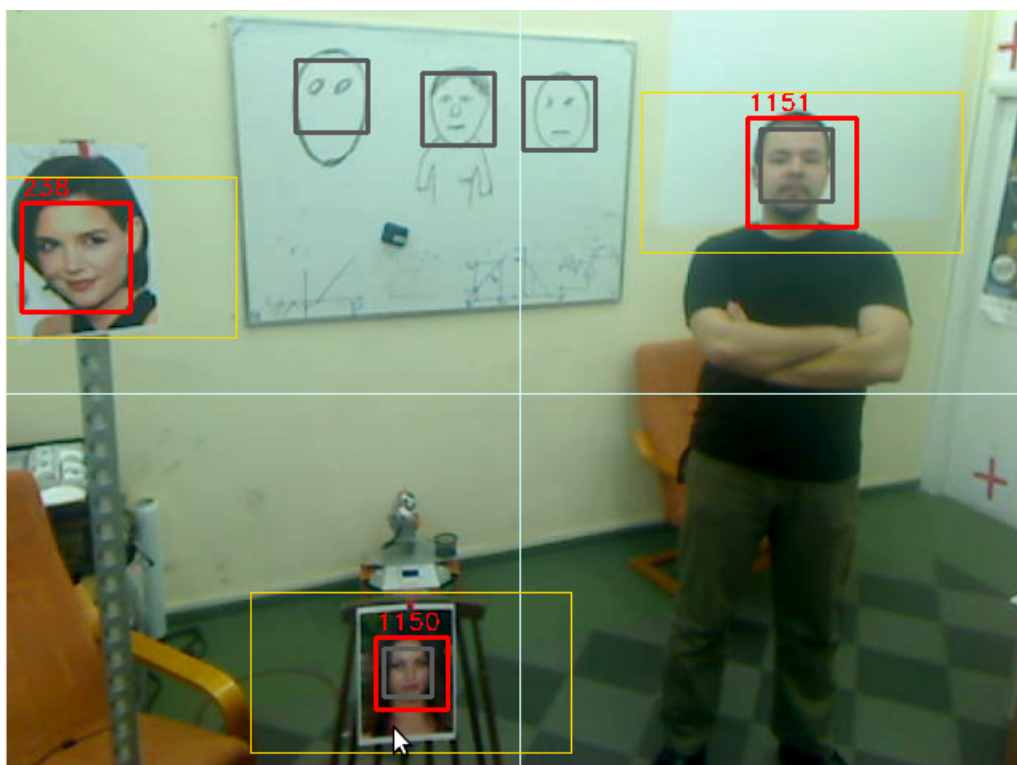
<sup>2</sup>Badania przeprowadzono w Laboratorium Robotów Społecznych (209) dzięki uprzejmości pracowników projektu LIREC.



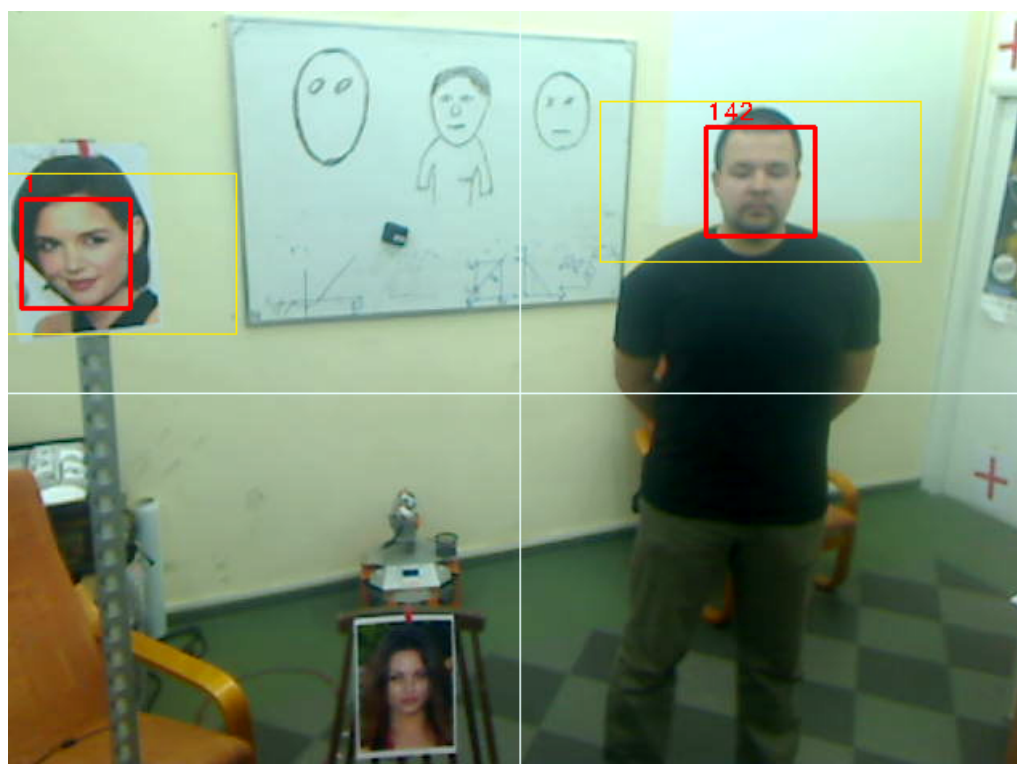
Rysunek 5.17 Przykład detekcji wielu twarzy (7).



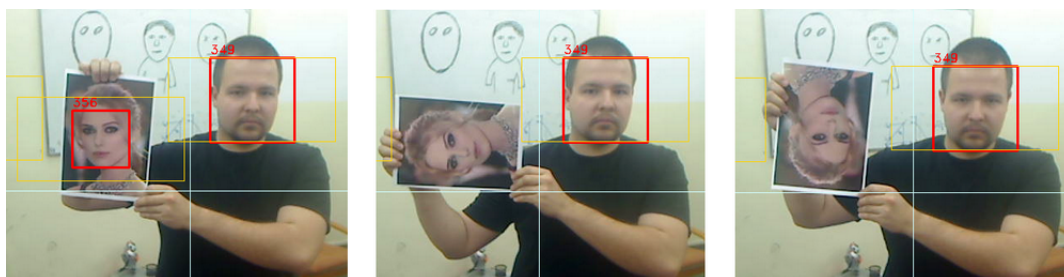
Rysunek 5.18 Przykład detekcji wielu twarzy (5).



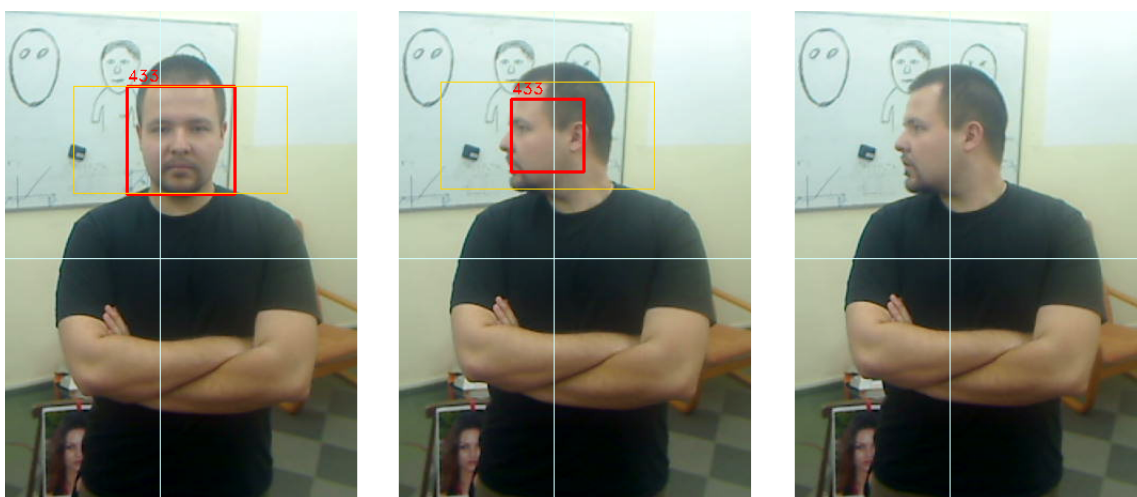
Rysunek 5.19 Przykład filtracji błędnie wykrywanych twarzy.



Rysunek 5.20 Przykład redukcji błędnie wykrywanych twarzy przez dobór parametrów.



Rysunek 5.21 Funkcja wykrywa jedynie twarze w orientacji pionowej.



Rysunek 5.22 Przykład utrzymywania ufności twarzy.

# Rozdział 6

## Podsumowanie

W ramach niniejszej pracy przygotowano głowicę wizyjną dla robota społecznego, umożliwiającą obrót w pionie i poziomie. Wykorzystane kamery ze zintegrowanym modulem *pan-tilt* wyposażono w kodery położenia niezbędne do zapewnienia wymaganej powtarzalności pozycjonowania i synchronizacji ruchów. Przeprowadzone badania dowodzą, jak ważne w sterowaniu jest sprzężenie zwrotne. Oprócz oczywistego zastosowania, jakim jest akwizycja obrazu, kamery stanowią dodatkową formę ekspresji emocji robota.

Dla głowicy zaprojektowano i wykonano sterownik, który wraz z regulatorami zapewnia sterowanie silnikami modułów *pan-tilt*. Oprócz 4 sygnałów PWM wymaganych przez głowicę wizyjną, sterownik zapewnia dodatkowe 8, co umożliwiło znaczne zredukowanie liczby elementów elektronicznych robota. Komunikacja ze sterownikiem odbywa się za pomocą interfejsu RS-485. Z myślą o projektach badawczych i dydaktycznych sterownik został wyposażony również w synchroniczny interfejs szeregowy SPI, interfejs zgodny z I2C oraz asynchroniczny interfejs szeregowy z możliwością handshakingu i poziomami napięć zgodnymi z RS-232C.

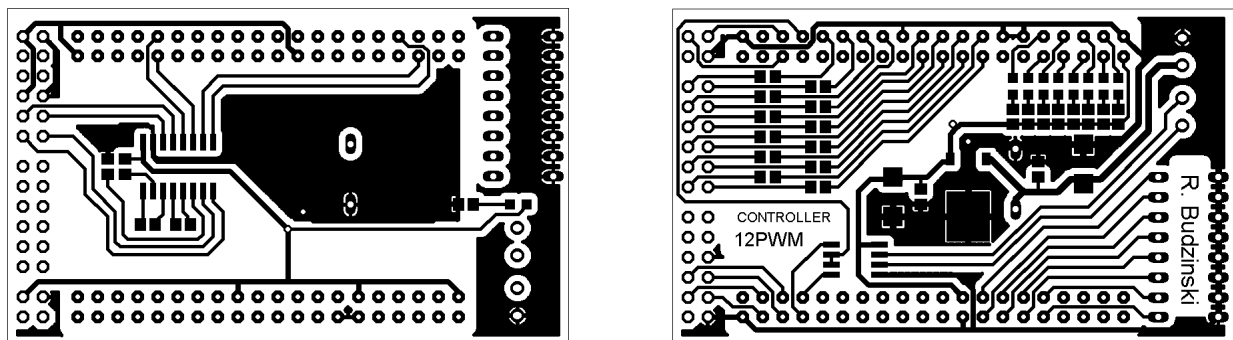
Przygotowany moduł detekcji twarzy wraz z zaimplementowanym filtrem odpowiada za wykrywanie i identyfikowanie twarzy oraz zwracanie informacji o ich położeniu. Ponadto uwzględnia orientację głowy oraz kamer zapisując twarze we współrzędnych w globalnym układzie odniesienia.

Opisywana głowica została wykorzystana w robocie społecznym Samuel, realizowanym przez Koło Naukowe Robotyków „KoNaR” z okazji obchodów 100-lecia Uczelni Technicznych we Wrocławiu. Robot witał gości przybywających na inaugurację obchodów 100-lecia w Operze Wrocławskiej oraz podczas Charytatywnego Balu Rektora Politechniki Wrocławskiej. O powodzeniu projektu może świadczyć 2. miejsce w kategorii freestyle na Międzynarodowych Zawodach Robotów w Wiedniu. Projekt budził ogromne zainteresowanie mediów. Robot został przedstawiony w programach: Kawa czy Herbata (TVP1), Dzień Dobry TVN (TVN), Fakty (TVN), Teleexpress (TVP1) oraz Fakty (TVP Wrocław).

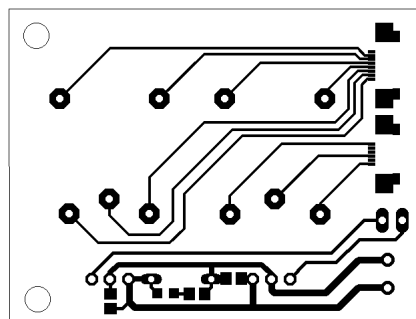
Szczegółowe informacje na temat robota społecznego Samuel znajdują się na oficjalnej stronie projektu <http://twingo.ict.pwr.wroc.pl/samuel>

# Dodatek A

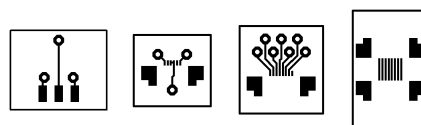
## Schematy



Rysunek A.1 Płytki PCB sterownika - warstwa górna i dolna.

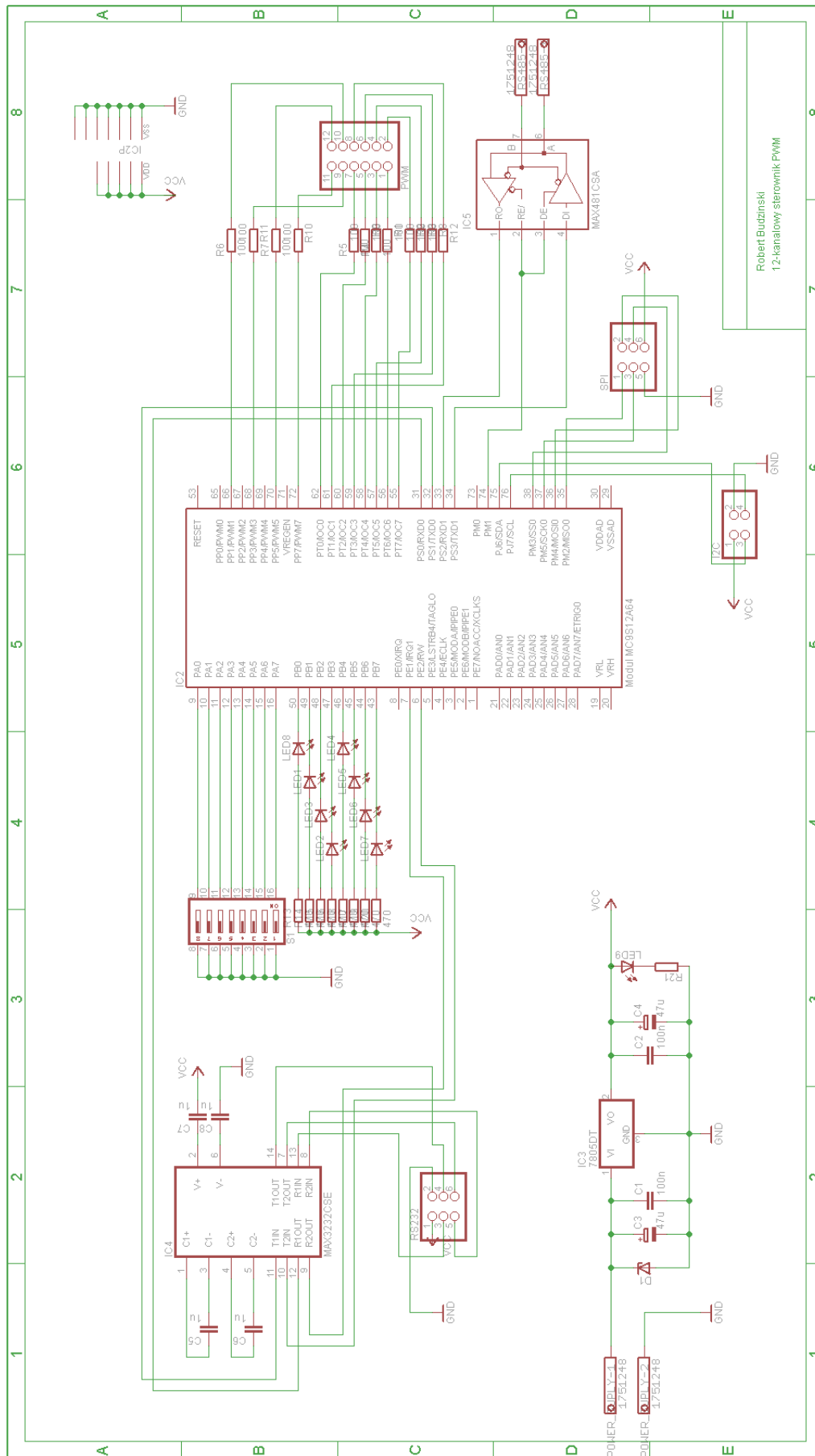


Rysunek A.2 Płytki PCB regulatora.

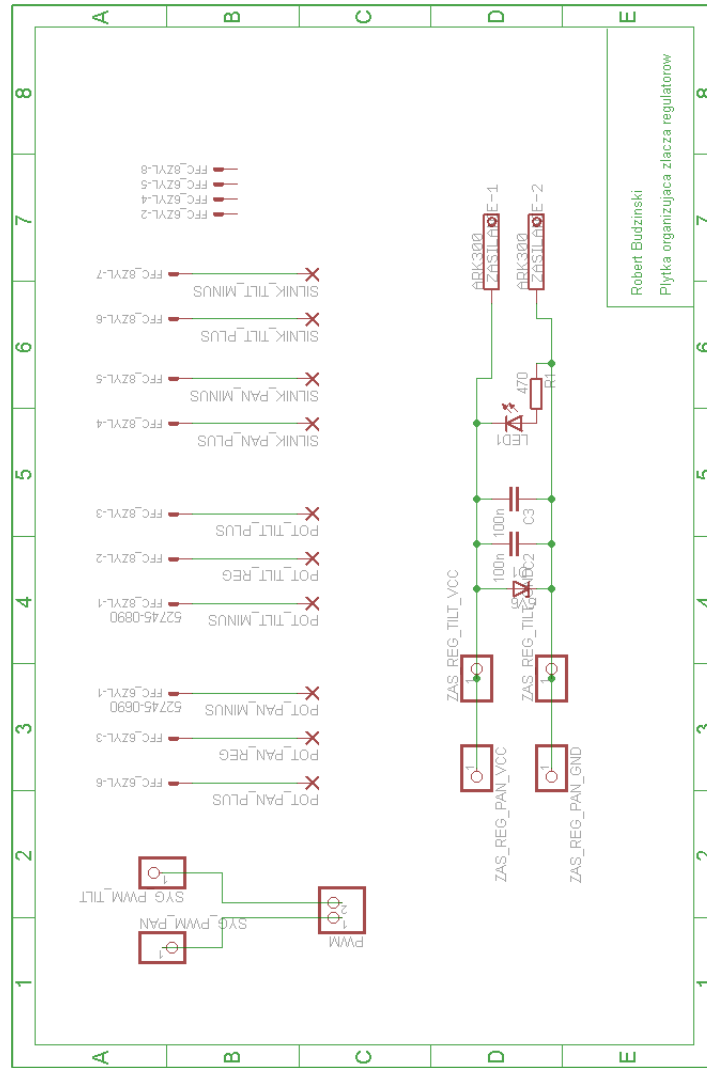


Rysunek A.3 Płytki połączeniowe dla potencjometrów i taśm FFC.





Rysunek A.4 Schemat ideowy sterownika.



Rysunek A.5 Schemat ideowy płytki regulatora.

# Dodatek B

## Opis złącz sygnałowych

złącze SVL				złącze SVR			
	1	2		VDD(+5V)	41	42	VSS(GND)
	3	4		PB7	43	44	PB6
	5	6	PE2/RW	PB5	45	46	PB4
PE1/IRQ	7	8		PB3	47	48	PB2
PA0	9	10	PA1	PB1	49	50	PB0
PA2	11	12	PA3	VDD(+5V)	51	52	VSS(GND)
PA4	13	14	PA5		53	54	VSS(GND)
PA6	15	16	PA7	PT7/IOC7	55	56	PT6/IOC6
VSS(GND)	17	18	VDD(+5V)	PT5/IOC5	57	58	PT4/IOC4
	19	20		PT3/IOC3	59	60	PT3/IOC2
	21	22		PT1/IOC1	61	62	PT0/IOC0
	23	24		VDD(+5V)	63	64	VSS(GND )
	25	26		PP0/PWM0	65	66	PP1/PWM1
	27	28		PP2/PWM2	67	68	PP3/PWM3
	29	30		PP4/PWM4	69	70	PP5/PWM5
PS0/RxD0	31	32	PS1/TxD0		71	72	PP7/PWM7
PS2/RxD1	33	34	PS3/TxD1		73	74	PM1
PM2/MISO0	35	36	PM4/MOSI0	PJ6/SDA	75	76	PJ7/SCL
PM5/SCK0	37	38	PM3/SS0		77	78	
VSS(GND)	39	40	VDD(+5V)		79	80	VSS(GND)

Tabela. B.1 Wykorzystane złącza modułu z układem MC9S12A64.

sygnał	wyprowadzenie	opis
PA0-PA7	9-16	przełącznik ustalający ID sterownika
PB0-PB7	50-43	sygnalizacyjne diody LED
PT0-PT7	55-62	wyjścia sygnałów PWM (generowane timerami)
PP1/PWM1	66	wyjścia sygnałów PWM (kanały)
PP3/PWM3	68	
PP5/PWM5	70	
PP7/PWM7	72	
PE1/IRQ	7	linia CTS (RS-232)
PE2/RW	6	linia RTS (RS-232)
PS0/RxD0	31	linia RX (RS-232)
PS1/TxD0	32	linia TX (RS-232)
PS2/RxD1	33	linia RX (RS-485)
PS3/TxD1	34	linia TX (RS-485)
PM1	74	linia DE, !RE (RS-485)
PM2/MISO0	35	linia MISO (SPI)
PM4/MOSI0	36	linia MOSI (SPI)
PM5/SCK0	37	linia SCK (SPI)
PM3/SS0	38	linia SS (SPI)
PJ6/SDA	75	linia SDA (IIC)
PJ7/SCL	76	linia SCL (IIC)
VDD(+5V)	18, 40, 41, 51, 63	zasilanie +5V
GND	17, 39, 42, 52, 54, 80	masa zasilania

Tabela. B.2 Wykorzystane sygnały modułu z układem MC9S12A64.

# Bibliografia

- [1] T. Fong, I. Nourbakhsh, K. Dautenhahn. *A survey of socially interactive robots: Concepts, design, and applications*. *Robotic and Autonomous Systems*, 42, 2003.
- [2] W. Wosińska, *Psychologia życia społecznego*. Gdańsk: Gdańskie Wydawnictwo Psychologiczne, 2004. ISBN 83-89574-34-9.
- [3] Kismet Home Page,  
<http://www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet.html>
- [4] L. Aryananda, J. Weber, *MERTZ: A Quest for a Robust and Scalable Active Vision Humanoid Head Robot*, MIT Computer Science and Artificial Intelligence Laboratory Cambridge, Massachusetts, 02139, USA.
- [5] Personal Robots Group  
<http://robotic.media.mit.edu>
- [6] *SwissRanger SR-3000 Manual*, Centre Suisse d'Electronique et de Microtechnique, 2006.
- [7] *Logitech QuickCam Sphere, User's Guide, 2007*.
- [8] *H.264 Mega-Pixel CMOS PT Internet Camera ICA-HM230 User's Manual*, Version 1.01, 2009.
- [9] *2.6x Pan-focus Zoom Progressive Scan CCD PZ7151/PZ7152 Network Camera User's Manual*
- [10] *Data Sheet AS5040 10Bit 360° Programmable Magnetic Rotary Encoder*, Revision 2.10, Austriamicrosystems AG, Revision 2.10, 2009.
- [11] *General Specification of HS-6975HB coreless digital servo*, Hitec RCD Korea Inc., 2003.
- [12] *AAS-313 / 13g Micro Servo (Ball Bearing)*, Version: 1.2, Alturn USA, LLC, 2008.
- [13] *Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers*, MAXIM, 1996.
- [14] R. Budziński, J. Kędzierski, B. Weselak, *Głowa robota Samuel – konstrukcja*, referat przyjęty na XI Krajową Konferencję Robotyki, Wrocław, 2010.
- [15] M. Wnuk, *Moduł z mikrokontrolerem MC9S12A64 (lub MC9S12C32)*, Raport IIAR PWr. serii SPR 11/2005, Wrocław, 2004.

- [16] *MC9S12DJ64 Device User Guide (Covers also MC9S12D64, MC9S12A64, MC9S12D32, MC9S12A32)* 9S12DJ64DGV1/D Rev. 01.20, Freescale Semiconductor, Inc., 2005.
- [17] R. Kuczaj, *Interfejsy OSBDM i TBDML*, Koło Naukowe Robotyków „KoNaR”, Politechnika Wrocławska, 03.2008.
- [18] *Dynamixel RX-64 - User's Manual*, ROBOTIS CO.,LTD.
- [19] S. Chlebicki, J. Kędzierski, M. Żarkowski, *System sterowania głową robota społecznego*, referat przyjęty na XI Krajową Konferencję Robotyki, Wrocław, 2010.
- [20] G. Bradski, A. Kaehler, *Learning OpenCV*, 2008, ISBN: 978-0-596-51613-0.
- [21] *A Guide to CodeWarrior Development Studio For HCS12(X) Microcontrollers*, Freescale Semiconductors, 2006.
- [22] *Processor Expert and Device Initialization User Guide*, UNIS, 2008.
- [23] *S12(X) Debugger Manual*, Freescale Semiconductors, 2009.
- [24] Intel Corporation, *Open Source Computer Vision Library*, 1999-2001.
- [25] Brian R. Duffy, *Anthropomorphism and Robotics*, Media Lab Europe, Sugar House Lane, Bellevue, Dublin 8, Ireland.