

Kamera internetowa z funkcją pomiaru temperatury i wilgotności **EriCAM v1.0**

Opracowanie:

Paweł Melnarowicz

1. Założenia projektu

Zadaniem projektowanego urządzenia jest wykonywanie zdjęć otoczenia i wysyłanie ich w postaci skompresowanej (w formacie JPEG) poprzez sieć ethernet. Urządzenie posiada interfejs do sieci ethernet oraz własny serwer WWW, za pośrednictwem którego zdjęcia są prezentowane w sieci. Odczyt odbywa się przy pomocy zwykłej przeglądarki internetowej bez konieczności używania specjalnego oprogramowania.

Dodatkową funkcję urządzenia jest pomiar temperatury oraz wilgotności powietrza i prezentacja tych danych na dynamicznie generowanej stronie WWW. Urządzenie jest zasilane za pośrednictwem sieci ethernet (wykorzystując pary przewodów standardowej skrętki UTP, które przy prędkości 10Mbit/s nie są użyte do transmisji danych). Urządzenie jest zamontowane w standardowej obudowie kamery kopułkowej o średnicy około 10cm co zapewnia estetyczny wygląd i prostotę montażu. Płaszczyznę pochylenia czujnika obrazu można zmieniać w niewielkich granicach co daje dużą swobodę wyboru miejsca zamontowania kamery.

2. Zastosowane elementy

Urządzenie zostało zbudowane w oparciu o moduł aparatu cyfrowego Ericsson MCA-25 CommuniCam (produkowany jako przystawka do starszych modeli telefonów komórkowych firmy Ericsson) oraz miniaturowy serwer WWW, którego sercem jest 8-bitowy mikrokontroler z rodziny AVR: Atmega128 wraz ze stosem TCP-IP udostępnianym na licencji Open Source przez firmę Egnite pod nazwą EtherNut lub Nut/OS.

Moduł MCA-25 ma postać niewielkiej płytki na której umieszczono kolorowy przetwornik obrazu wraz z obiektywem o rozdzielczości 640x480 pixeli, procesor zgodny z ARM oraz pamięć DRAM i Flash. Moduł komunikuje się z współpracującym układem poprzez interfejs UART dzięki czemu podłączenie go do mikrokontrolera Atmega nie sprawia żadnego problemu. Po wysłaniu odpowiedniej komendy sterującej moduł wykonuje zdjęcie, kompresuje je do formatu JPEG i następnie przesyła poprzez łącze szeregowe. Cała ta operacja trwa około kilku sekund w zależności od objętości wykonanego zdjęcia. Protokół komunikacji z modułem oraz przykładowe oprogramowanie w języku C można znaleźć w sieci Internet.

Rolę serwera WWW pełni popularny mikrokontroler firmy Atmel - Atmega128. Jest on taktowany zegarem o częstotliwości 16MHz a większość z jego instrukcji jest wykonywana w pojedynczym cyklu zegarowym dzięki czemu jest on wystarczająco szybki aby sprawnie przysyłać dane poprzez ethernet. Wewnętrzna pamięć programu procesora jest zbyt mała aby uruchomić stos TCP-IP EtherNut dlatego dołączono do niego zewnętrzną pamięć SRAM.

Interfejs do sieci ethernet oraz warstwa fizyczna stosu została zrealizowana w oparciu o popularny układ RTL8019AS firmy Realtek. Wybrano ten właśnie układ ze względu na możliwość pracy w trybie 8-bitowym, jego obsługę przez system EtherNut oraz stosunkowo niską cenę.

Do pomiaru temperatury i wilgotności użyty został układ SHT-11 firmy Sensiron. Układ ten wykonany w innowacyjnej technologii CMOSens, która pozwala na integrację czujnika wilgotności bezpośrednio na strukturze krzemowej układu. Dzięki temu w jednej strukturze zmieszczono sensor wilgotności, temperatury oraz 15-bitowe przetworniki analogowo-cyfrowe i źródła referencyjne. Sensor jest kalibrowany fabrycznie dzięki czemu można uniknąć tej kłopotliwej czynności w trakcie uruchamiania urządzenia. Jest zasilany

napięciem od 3.3V do 5V i komunikują się z współpracującym mikrokontrolerem przy pomocy magistrali 2-przewodowej o standardzie zbliżonym do I²C.

3. Ograniczenia

Wybór modułu MCA-25 i serwera na procesorze 8-bitowym daje korzyści w postaci prostoty układu oraz jego niskiej ceny i możliwości wykonania w warunkach domowych z podzespołów dostępnych na krajowym rynku. Niestety niesie to za sobą pewne ograniczenia:

- kompresja obrazu do formatu JPEG jest bardzo złożonym procesem i wymaga sporej mocy obliczeniowej. Procesor ARM taktowany zegarem 96MHz zastosowany w module MCA-25 radzi sobie z kompresją i przesyłem danych w około 7 sekund. Widać więc, że kamera nadaje się do wykonywania statycznych zdjęć z częstotliwością praktycznie nie większą niż 1 klatka na 10 sekund.
- Moduł MCA-25 był przewidziany jako przystawka do telefonu komórkowego i nie został wyposażony w funkcje sterowania obrazem (nie można zmieniać kontrastu ani balansu bieli itp.).
- Przetwornik MCA-25 wymaga dość dobrego oświetlenia aby wykonywane zdjęcia były dobrej jakości.
- Zastosowany serwer WWW ma możliwość jednoczesnej obsługi 4 połączeń HTTP, lecz zastosowana pamięć jest zbyt mała aby buforować zdjęcia w związku z czym tylko jedno połączenie w danej chwili może odbierać obraz.

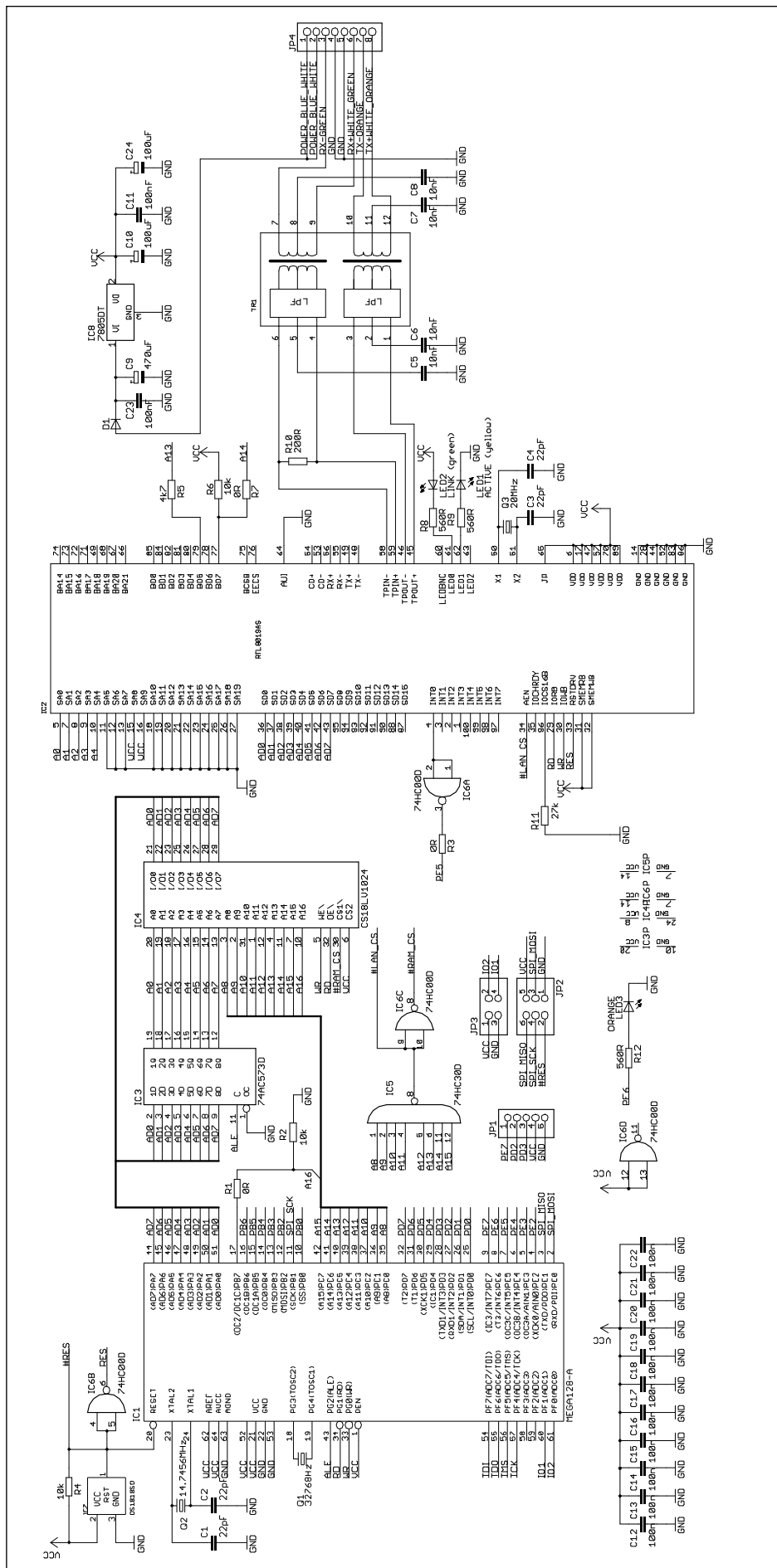
Ostatnie ograniczenie można w prosty sposób ominąć stosując dodatkowe oprogramowanie np. na routerze (serwer proxy lub dedykowany program buforujący).

4. Zastosowania

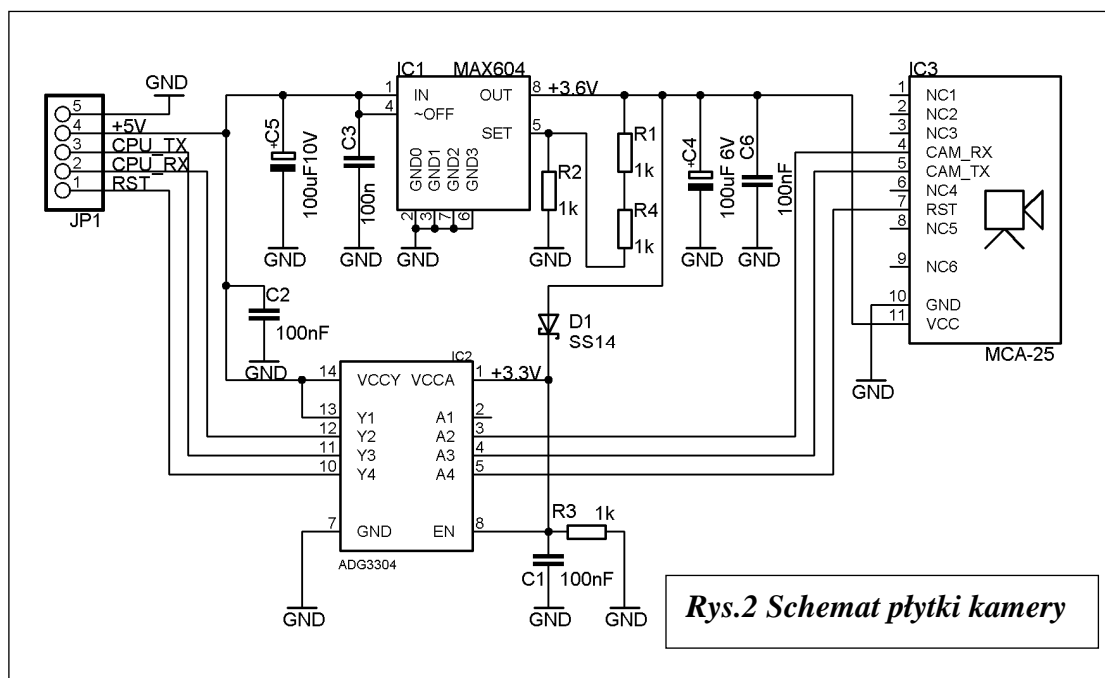
Projektowane urządzenie może znaleźć zastosowanie wszędzie tam, gdzie potrzebne jest okresowe wykonywanie zdjęć jakiegoś obiektu z niewielką częstotliwością. Może być na przykład elementem stacji meteorologicznej (będzie pokazywać co naprawdę dzieje się za oknem) lub służyć jako element niskobudżetowego systemu nadzoru (np. w odlegle położonym domku letniskowym gdzie jako medium komunikacyjne wykorzystuje się transmisję po sieci GSM o bardzo małej przepustowości więc i tak nie można przesłać zdjęć szybciej niż umożliwia to omawiana kamera).

5. Schemat elektryczny

Urządzenie zostało zbudowane z dwóch współpracujących ze sobą części - głównej płytki serwera oraz płytki konwertera poziomów.



Rys .1 Sch emat płytki baz owe j



Rys.2 Schemat płytki kamery

5.1 Płytki serwera

Schemat płytki serwera przedstawiony został na rysunku 1. Jak widać, sercem urządzenia jest układ IC1 - Mikroprocesor ATmega128. Kwarce 14.7456MHz służy do generowania sygnału taktującego mikrokontroler, częstotliwość ta została wybrana tak, by po podzieleniu w wewnętrznym preskalerze otrzymać w prosty sposób dokładną częstotliwość taktowania portów szeregowych (UART0 - 115200bps do debuggingu i UART1 do obsługi kamery).

Do mikroprocesora, za pośrednictwem zatrzaśku IC3 została podłączona pamięć zewnętrzna SRAM o pojemności 128kB - jest to układ IC4. Procesor ATmega128 sprzętowo obsługuje jedynie 64kB pamięci. Linia adresowa A16 może być zwarta do masy zasilania lub podłączona do portu PB7 w zależności, który z rezystorów (R1 czy R2) zostanie wlutowany. Podłączenie do pinu PB7 daje możliwość programowej obsługi bankowania pamięci i wykorzystania jej w pełni. W użytej wersji systemu Nut/OS opcja ta nie jest jednak dostępna. Układy IC5 oraz IC6C spełniają rolę prostego dekodera adresowego służącego do rozgraniczenia obszaru adresowego zajmowanego przez pamięć i kontroler ethernet. Przy takim połączeniu adres bazowy kontrolera sieci wynosi 0xFF00, a rozmiar pamięci dostępnej w programie to 65280 bajtów (prawie 64kB).

Rolę kontrolera Ethernet spełnia układ IC2 i jest on połączony zgodnie ze standardowym schematem aplikacyjnym. Rezystory R5, R6 i R7 służą do wyboru trybu emulacji pamięci EEPROM. Układ RTL8019AS odczytuje pewne dane konfiguracyjne (np. adres MAC) z pamięci szeregowej EEPROM, która powinna być do niego podłączona. W omawianym układzie pamięć ta jest emulowana przez procesor, o ile wlutowane zostaną rezystory R5 i R7. Jeśli opcja ta nie jest wykorzystana w programie, należy wlutować rezystor R6.

Diody LED1 i LED2 służą do wizualnej kontroli działania interfejsu. Dioda LED2 (zielona) świadczy o poprawnym skonfigurowaniu interfejsu i nawiązaniu połączenia, natomiast dioda żółta LED1 zapala się w trakcie przesyłu danych. Układ ethernet może działać zależnie od sterownika w trybie pollingu (procesor cyklicznie sprawdza czy pojawiły

się dane) lub używając przerwań procesora. Jeśli wykorzystana jest ta druga opcja (polecane i domyślnie ustawione w konfiguracji Nut/OS), należy wlutować rezystor R3. Izolację galwaniczną pomiędzy siecią ethernet (standardowa skrętka) a układem zapewnia transformator TR1.

Układ kamery przystosowany jest do zasilania napięciem stałym niestabilizowanym o wartości około 8V. Większe napięcie nie jest wskazane ze względu na trudności z odprowadzeniem ciepła ze stabilizatora IC8. Pobór prądu przez układ kamery może sięgać nawet 500mA (w impulsach, np. w trakcie wykonywania zdjęć).

Mikroprocesor i układ kontrolera sieci wymagają doprowadzenia impulsu zerującego po włączeniu i ustabilizowaniu się zasilania. Do jego generowania zastosowano specjalizowany układ firmy Maxim - DS1818. W przypadku trudności z jego zakupem można zamiast niego wlutować kondensator o pojemności około 330nF pomiędzy wyprowadzenia 1 i 3 lecz jest to rozwiązanie bardzo niepewne i nie polecane.

Kondensatory C12-C22 służą do odsprężania zasilania i muszą być zamontowane blisko odpowiednich wyprowadzeń układów scalonych.

Złącze JP1 służy do podłączenia współpracującej płytki z kamerą. Złącze JP3 służy do ewentualnej rozbudowy urządzenia. Przykładem takiej rozbudowy jest dołączenie czujnika temperatury i wilgotności SHT11. Czujnik należy dołączyć bezpośrednio do złącza: pin VCC do nóżki Vdd, pin IO1 do nóżki DATA, pin IO2 do nóżki CLK a pin GND do nóżki GND.

Złącze JP2 służy do wgrywania oprogramowania do pamięci Flash mikrokontrolera. Złącze to ma układ pinów zgodny ze standardem STK200 i taki właśnie programator należy użyć do programowania. Schemat programatora można łatwo znaleźć w sieci Internet lub kupić gotowy moduł podłączany do portu LPT komputera.

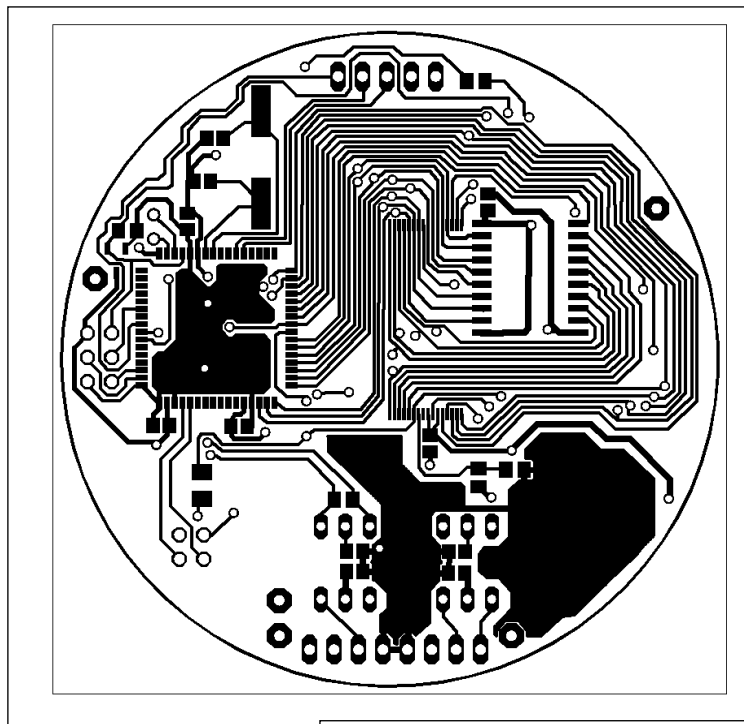
5.2 Płytki kamery

Schemat płytki z kamerą został przedstawiony na **rysunku 2**. Zadaniem tego układu jest zapewnienie stabilnego zasilania kamery oraz konwersja poziomów sygnałów sterujących. Moduł MCA-25 wymaga zasilania napięciem o wartości 3.6V - 3.7V. Napięcie to powinno być dobrze odfiltrowane, ponieważ jak pokazały próby, ma to wpływ na jakość otrzymywanych zdjęć. Linie sterujące (TxD, RxD i RST) powinny pracować w standardzie 3.3V.

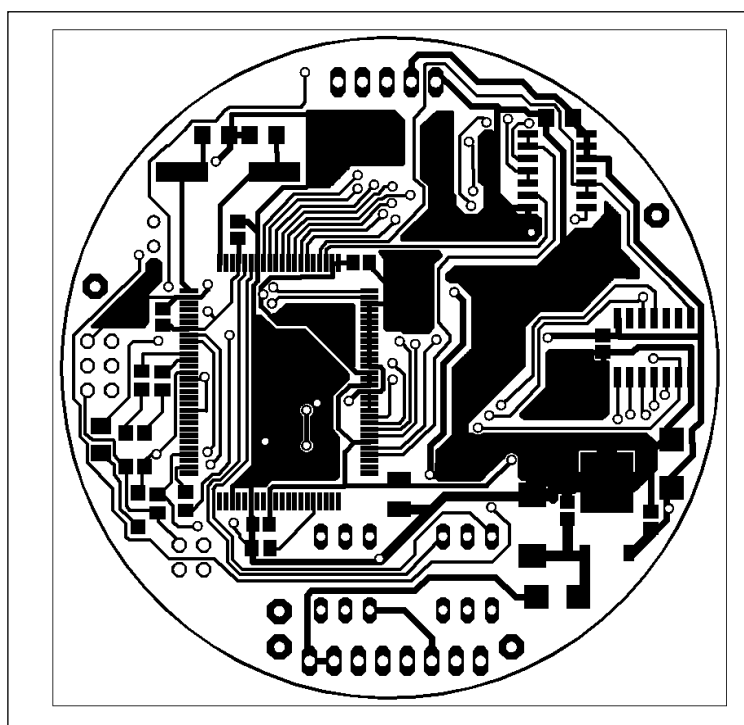
Układ IC1 - MAX604-ADJ jest miniaturowym stabilizatorem LDO (Low Drop) o wartości napięcia wyjściowego ustalonej przy pomocy dzielnika rezystorowego. Elementy R1, R2 i R4 decydują o wartości napięcia wyjściowego dlatego powinny to być rezystory metalizowane o tolerancji 1%. Ich wartość powinna wynosić od 1kΩ do 100kΩ (ważne jest by były jednakowe). Dzielnik ten powoduje ustalenie napięcia wyjściowego na poziomie 3 razy większym niż wewnętrzne napięcie odniesienia układu stabilizatora czyli dokładnie 3,6V. Napięciem tym jest zasilana kamera MCA-25 oraz za pośrednictwem diody D1 układ IC2 - ADG3304. Układ ten jest dwukierunkowym konwerterem poziomów z samoczynnym wykrywaniem kierunku transmisji na każdym z pinów produkowanym przez Analog Devices. Jego zadaniem jest konwersja sygnałów logicznych o amplitudzie 5V właściwych dla mikrokontrolera do standardu o amplitudzie 3.3V i na odwrót. Napięcie 3.3V jest uzyskiwane z napięcia 3.6V przy pomocy diody Schottky'ego D1 oraz rezystora obciążającego R3. Moduł kamery MCA-25 jest dołączony do układu przy pomocy oryginalnego złącza z telefonu komórkowego Ericsson T68i.

6.Montaż

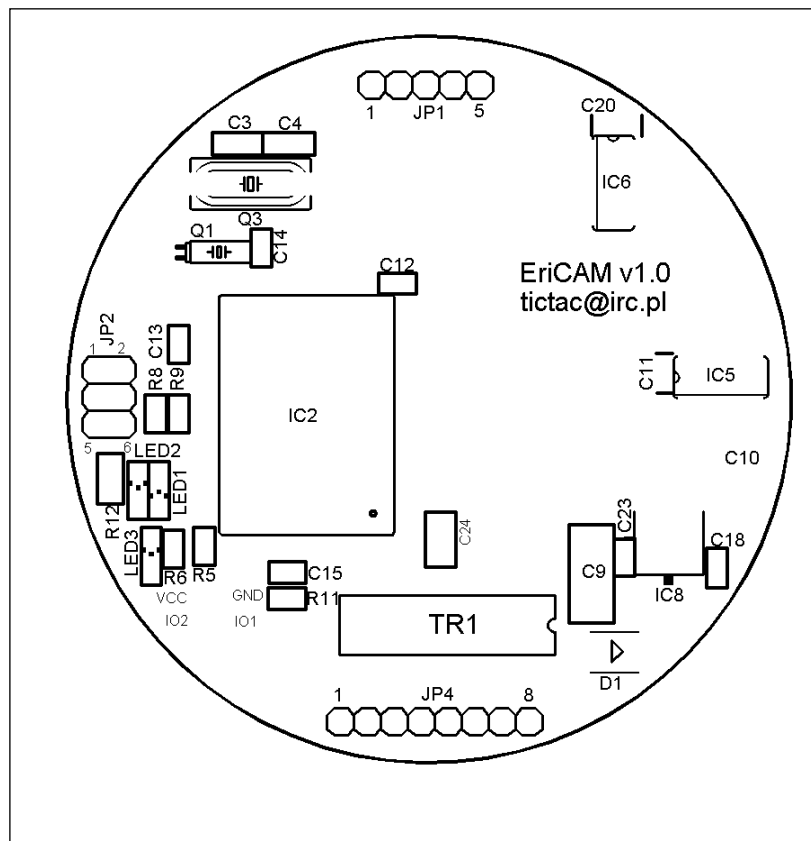
Urządzenie składa się z dwóch dwuwarstwowych płytek drukowanych, których mozaiki ścieżek oraz rozmieszczenie elementów przedstawiono na kolejnych rysunkach.



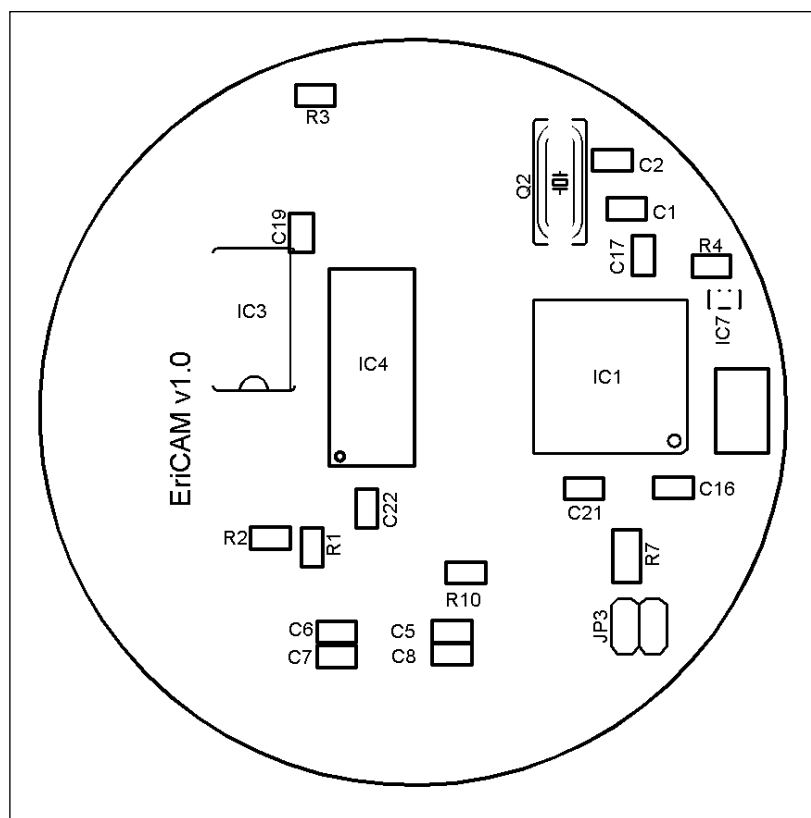
Rys.3 Warstwa miedzi - dół



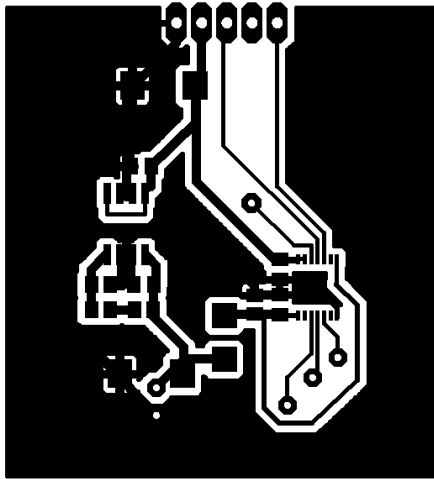
Rys.4 Warstwa miedzi - góra



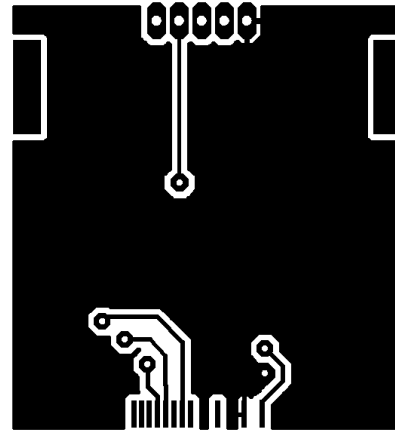
Rys.5 Rozmieszczenie elementów - góra



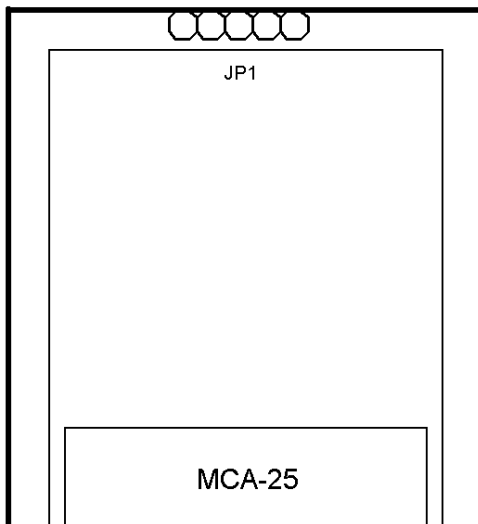
Rys.6 Rozmieszczenie elementów - dół



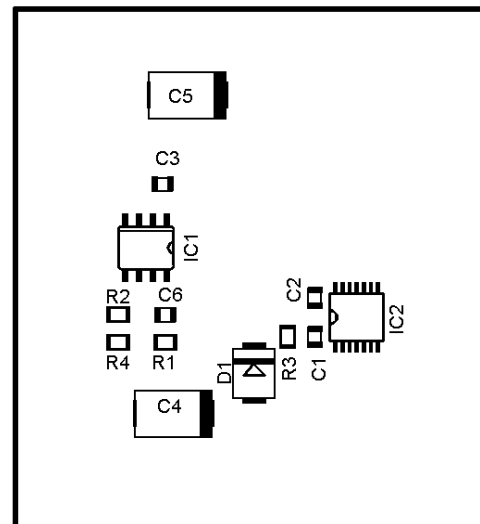
Rys.7 Warstwa miedzi - dół



Rys.8 Warstwa miedzi - góra



Rys.9 Rozmieszczenie elementów - strona górna



Rys.10 Rozmieszczenie elementów - strona dolna

Po zlutowaniu elementów obydwu płytek należy przygotować kabel i złącze JP4. Połączenia pomiędzy pinami złącza i kolorami przewodów typowej skrętki UTP przedstawia poniższa tabela:

Pin złącza JP4	Kolor przewodu
1	Niebieski (PoE+)
2	Niebiesko-biały (PoE+)
3	Zielony
4	Brązowy (PoE-)
5	Brązowo-biały (PoE-)
6	Zielono-biały
7	Pomarańczowy
8	Pomarańczowo-biały

Połączenie to jest typu prostego i tak przygotowany kabel nadaje się do podpięcia do switch'a lub routera poprzez zasilacz PoE (Power Over Ethernet) który powinien dostarczać napięcie około 8V i mieć wydajność prądową większą niż 500mA.

Jeżeli opcja zasilania przez ethernet nie będzie wykorzystana, zasilanie należy doprowadzić do pinów 1/2 złącza JP4 (+8V) oraz 4/5 (masa).

Po sprawdzeniu poprawności montażu do płytki procesora można podłączyć zasilanie oraz programator STK-200 i zaprogramować mikrokontroler. Po pomyślnym wykonaniu tej czynności można połączyć ze sobą elektrycznie i mechanicznie obie płytki. Poprawne działanie kamery sygnalizowane jest świeceniem diod LED. Dioda zielona świeci, gdy aktywne jest połączenie ethernet, dioda żółta gdy przesyłane są dane, natomiast dioda czerwona zapala się w kilka sekund po włączeniu zasilania i sygnalizuje poprawną inicjalizację modułu MCA-25 oraz gotowość do wykonywania zdjęć.

7. Oprogramowanie

7.1 System operacyjny

Oprogramowanie kamery zostało oparte o system Nut/OS 4.3 w wersji dla mikrokontrolerów Atmel/AVR. System ten jest dostępny w wersji źródłowej na stronie internetowej firmy Egnite oraz na oficjalnej stronie systemu - www.ethernut.de. Do systemu dołączona jest obszerna dokumentacja.

Oprogramowanie Nut/OS składa się z drzewa katalogów zawierających pliki źródłowe z implementacją poszczególnych funkcji jądra systemu, bibliotekami obsługi protokołów sieciowych oraz sterownikami urządzeń. Do systemu dołączony jest konfigurator pozwalający na ustalenie parametrów platformy sprzętowej, na której uruchomiony zostanie Nut/OS. W niniejszym projekcie konfigurator ten nie został wykorzystany a wszystkie pliki konfiguracyjne edytowane były ręcznie. Dodatkowo pliki Makefile wspomagające kompilację systemu zostały przepisane od nowa i dostosowane do specyficznych potrzeb projektu. Z wielu plików źródłowych wchodzących w skład systemu Nut/OS wybrane zostały tylko te niezbędne do działania urządzenia i tylko one podlegać będą kompilacji. Takie podejście jest

dużo bardziej przejrzyste dla kogoś, kto pierwszy raz styka się z systemem Nut/OS i ogranicza możliwość powstania trudnych do rozwikłania błędów kompilacji i linkowania.

7.2 Wymagane oprogramowanie

Do uruchomienia kamery potrzebne jest następujące oprogramowanie:

- Archiwum z systemem Ethernut - Nut/OS zmodyfikowanym dla potrzeb zastosowania w kamerze oraz aplikacja webserver.
- Kompilator języka C dla mikrokontrolerów AVR - do kompilacji oprogramowania prototypu użyty został kompilator avr-gcc w wersji 3.6 dla systemu Linux. Sprawdzono, że kompilacja odbywa się poprawnie także dla wersji pracującej w środowisku Windows - pakiet WinAVR 20070122 (wersja 4.1).
- W przypadku pracy w środowisku Windows wygodny może okazać się pakiet Cygwin pozwalający na korzystanie z Unix'owego shella i jego poleceń.
- Programator obsługujący interfejs STK-200. Do programowania prototypu użyty został program UISP 1.0.6.0 (działa w środowisku Linux lub Windows po zainstalowaniu bibliotek pakietu Cygwin oraz sterownika bezpośredniego dostępu do portu LPT o nazwie giveio.sys). W środowisku Windows można także skorzystać z graficznego narzędzia PonyProg2000 2.07 ustawiając typ interfejsu programującego na STK-200/STK-300.

7.3 Struktura systemu

Po rozpakowaniu archiwum z systemem Nut/OS na dysku pojawia się katalog Ethernut z następującą zawartością:

app - katalog zawierający aplikacje użytkownika

arch - katalog zawierający pliki zależne od architektury

crt - katalog z implementacją biblioteki C runtime w wersji zgodnej z systemem Nut/OS

dev - katalog zawierający implementacje sterowników urządzeń

fs - katalog zawierający implementacje bibliotek obsługi systemu plików

include - katalog z plikami nagłówkowymi

lib - w tym katalogu po kompilacji pojawiają się biblioteki systemowe

net - katalog z implementacją funkcji sieciowych

os - katalog z implementacją jądra systemu

pro - katalog z implementacją protokołów sieciowych wyższego rzędu

tools - źródła programu cruom służącego do kompilacji systemu plików pnutils.

Makedefs - plik zawierający definicje dla programu Make - zapisane są w nim najważniejsze opcje kompilacji

Makefile - plik poleceń narzędzia Make służącego do wspomagania kompilacji

Makerules - zbiór reguł narzędzia Make

7.4 Kompilacja systemu

Kompilacja systemu polega na wpisaniu polecenia make znajdując się w głównym katalogu systemu. Jeśli proces kompilacji (który trwa około minuty na typowym komputerze PC) zakończy się pomyślnie, tworzone są biblioteki systemowe.

7.5 Konfiguracja i kompilacja aplikacji

Następnym krokiem jest kompilacja konkretnej aplikacji. Aplikacja obsługująca kamerę i serwer www znajduje się w katalogu app pod nazwą webcam_meteo.

Aby ją skompilować należy przejść do tego katalogu i wykonać polecenie make. Wynikiem jego działania jest utworzenie między innymi pliku httpserv.hex. Jest to plik binarny w formacie Intel Hex przeznaczony do zaprogramowania pamięci flash mikrokontrolera.

Przedtem warto jednak przejrzeć plik httpserv.c który jest głównym plikiem aplikacji. Można w nim zmienić adres MAC interfejsu sieciowego oraz adres IP i maskę podsieci.

W tym celu należy edytować następujące linie:

```
#define MYMAC 0x00, 0x06, 0x98, 0x00, 0x00, 0x00
#define MYIP „192.168.1.77”
#define MYMASK „255.255.255.0”
```

Dostęp do kamery może być zabezpieczony hasłem (hasło jest przesyłane jawnym tekstem bez szyfrowania). Mechanizm ten można wyłączyć komentując linię 465 w pliku httpserv.c: NutRegisterAuth(„cgi-bin”, „tictac:tajne”);

W tej samej linii można zmienić nazwę użytkownika uprawnionego do przeglądania zdjęć oraz jego hasło. W podanym przykładzie jest to użytkownik „tictac” i hasło „tajne”.

7.5 Programowanie bitów „fuses” procesora

Mikroprocesory Atmega128 posiadają w obszarze pamięci flash specjalne rejestry służące do konfiguracji i zabezpieczenia kodu. Po zakupie procesor jest skonfigurowany do działania z wewnętrznym generatorem zegarowym oraz w trybie kompatybilności z procesorem Atmega103. Aby uruchomić system Nut/OS należy zmodyfikować wartości odpowiednich bitów aby procesor działał z pełnią swoich możliwości. W tym celu należy ustawić wartość słów konfiguracyjnych w sposób następujący: (zakładając użycie programatora UISP)

```
# uisp -dprog=stk200 -dpart=atmega128
--wr_fuse_l=0x3f --wr_fuse_h=0x83 --wr_fuse_e=0xff
```

Odpowiedź programatora powinna być zbliżona do poniższej:

```
Firmware Version: 49.48
Atmel AVR ATmega128 is found.
Fuse Low Byte set to 0x3f
Fuse High Byte set to 0x83
Fuse Extended Byte set to 0xff
```

W przypadku użycia innego programatora należy postąpić podobnie kierując się dołączoną dokumentacją. Operacja ta powinna być wykonana bardzo ostrożnie ponieważ modyfikacja nieodpowiedniego bitu może zablokować dostęp do pamięci lub wyłączyć interfejs ISP co powoduje że procesor staje się praktycznie bezużyteczny i konieczna jest jego wymiana.

7.6 Programowanie mikroprocesora

Programowanie odbywa się w sposób zależny od posiadanej aplikacji programującej: w środowisku Linux mając do dyspozycji program UISP wystarczy wydać polecenie „make program” i odczekać aż oprogramowanie zostanie przesłane poprzez interfejs ISP. W przypadku użycia PonyProg2000 należy postępować zgodnie z instrukcjami pojawiającymi się na ekranie i z dokumentacją do tego programu.

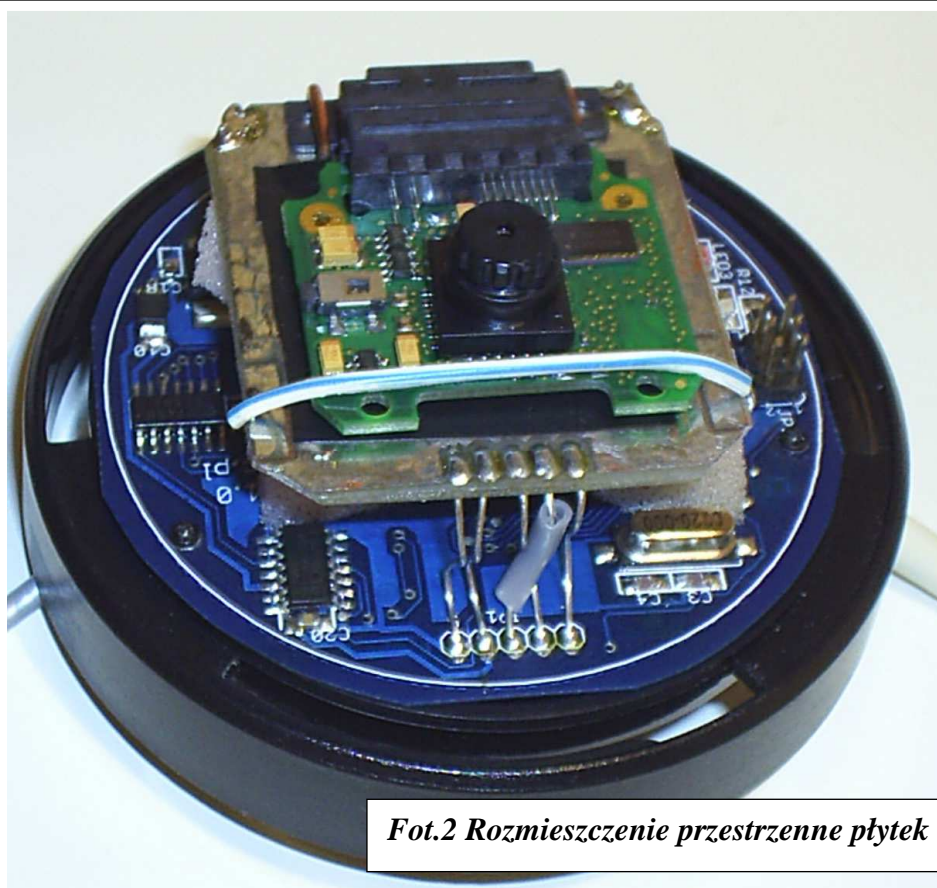
8. Montaż mechaniczny

Płytki drukowane zostały zaprojektowane w sposób umożliwiający połączenie ich przy pomocy drutu - srebrzanki tak, aby jedna znajdowała się nad drugą. Wzajemne pochylenie można zmieniać wyginając drut dostosowując płaszczyznę pochylenia kamery. Zespół dwóch płytek nadaje się do umieszczenia w typowej obudowie kamery kopułkowej dostępnej w handlu. Czujnik temperatury należy umieścić na zewnątrz obudowy dbając aby długość przewodu połączeniowego nie przekroczyła około 30cm. Do połączenia czujnika z płytą najlepiej jest użyć 4-żyłowego przewodu ekranowanego (ekran łącząc z wyprowadzeniem GND złącza). Czujnik SHT-11 jest bardzo delikatny i należy zabezpieczyć go poprzez umieszczenie w koszulce termokurczliwej w sposób zapewniający łatwy dopływ powietrza do okienka na jego obudowie.

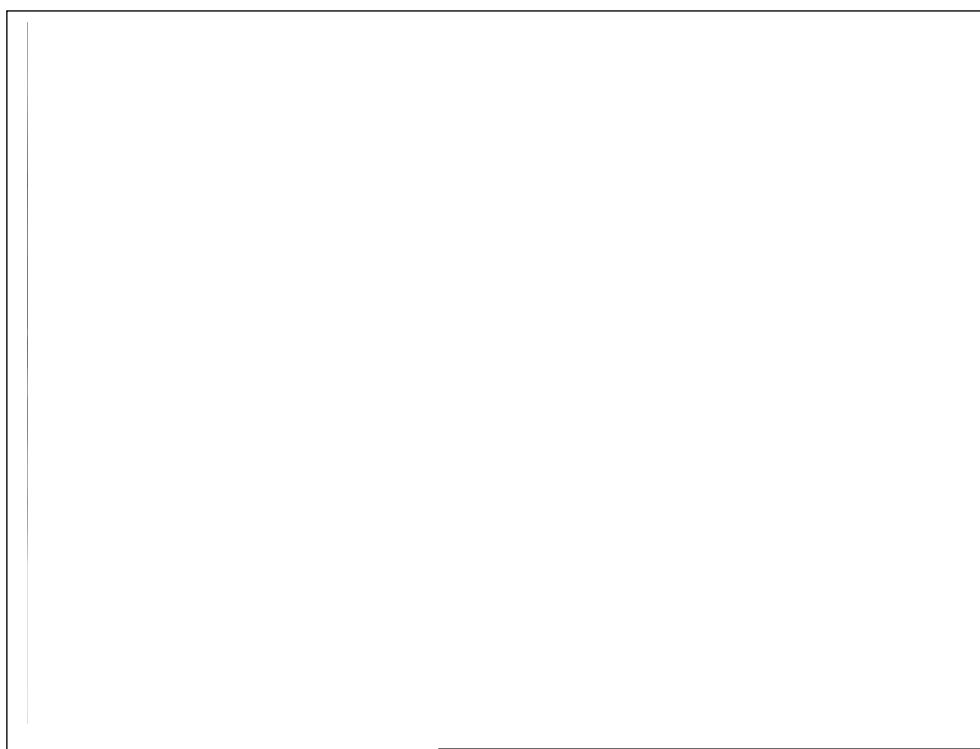
Szczegóły montażu ilustrują zdjęcia:



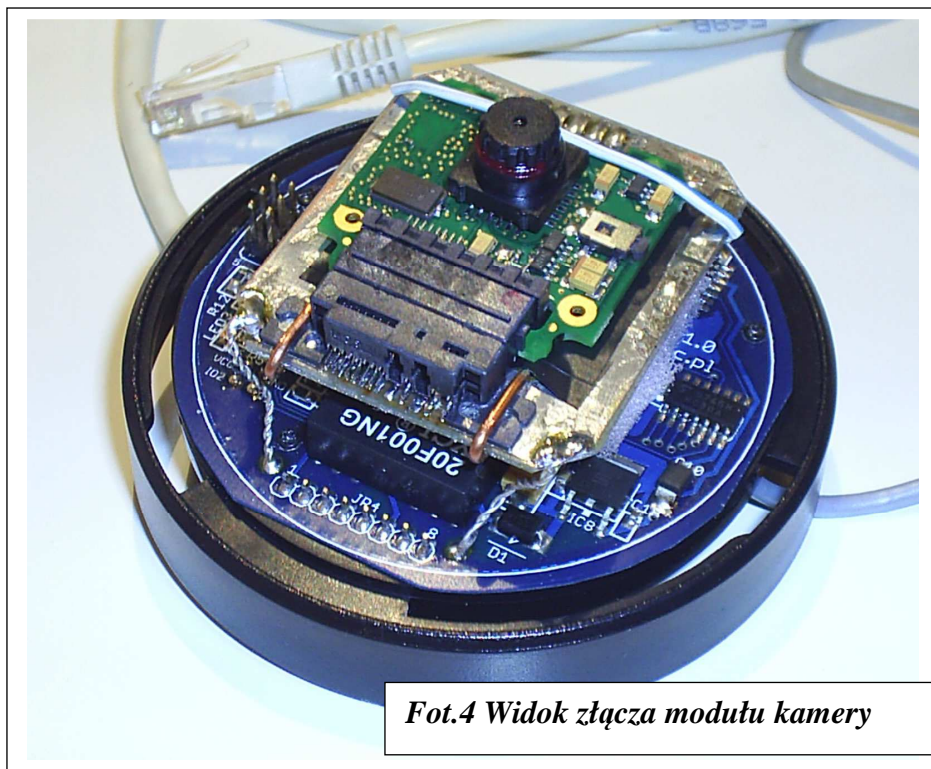
Fot.1 Izolacja czujnika SHT-11



Fot.2 Rozmieszczenie przestrzenne płytek

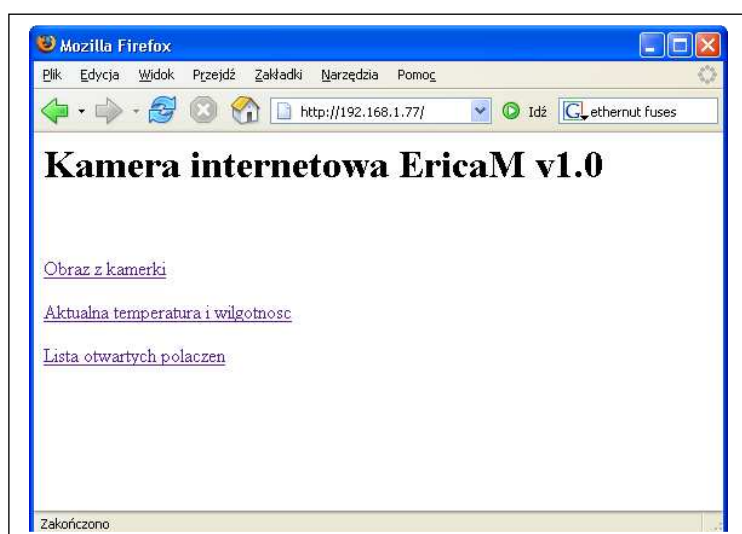


Fot.3 Kamera w obudowie



9. Obsługa kamery

Obsługa kamery jest bardzo prosta - po podłączeniu do sieci (nie zapominając o zasilaczu PoE) wewnątrz obudowy powinny zapalić się odpowiednie diody. Po odczekaniu kilku sekund aż zapali się dioda czerwona można uruchomić przeglądarkę WWW na dowolnym komputerze w podsieci i wpisać w jej pole adresu adres IP kamery (np. <http://192.168.1.77>). Na ekranie powinna pojawić się prosta strona WWW:



Rys.11 Strona główna serwera www

Po kliknięciu na link „*Obraz z kamery*” i podaniu poprawnej nazwy użytkownika i hasła, po kilku sekundach powinna pojawić się strona zawierająca zdjęcie sceny widzianej przez kamerę.

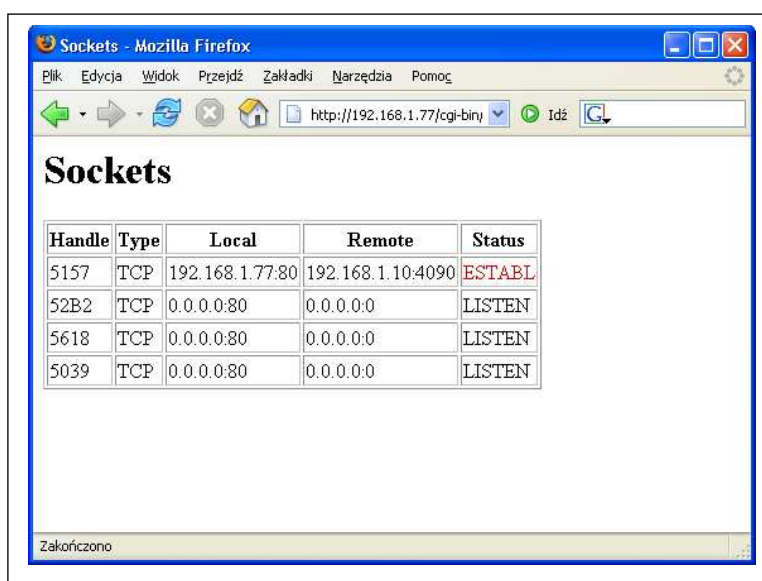
Po kliknięciu na link „*Aktualna temperatura i wilgotność*” pojawia się strona zawierająca informację o temperaturze i wilgotności. Dane są aktualizowane po każdym odświeżeniu strony. Częste odświeżanie (np. co sekundę) nie jest polecane gdyż czujnik ma bardzo małą bezwładność i przepływ prądu przez niego samego przy zbyt częstych pomiarach może powodować lekkie zawyżenie wskazań temperatury.

Link „*Lista otwartych połączeń*” daje dostęp do strony zawierającej listę aktualnie połączonych użytkowników i stan 4 nasłuchujących procesów serwera.

Przykładowe zrzuty ekranu znajdują się na kolejnych ilustracjach:



Rys.12 Strona informacji o temperaturze i wilgotności



Rys.13 Strona diagnostyczna



Fot.5 Przykładowe zdjęcie wykonane przez kamerę

Spis treści

1. Założenia projektu	2
2. Zastosowane elementy	2
3. Ograniczenia	3
4. Zastosowania	3
5. Schemat elektryczny	3
5.1 Płytki serwera	5
5.2 Płytki kamery	6
6. Montaż	7
7. Oprogramowanie	10
7.1 System operacyjny	10
7.2 Wymagane oprogramowanie	11
7.3 Struktura systemu	11
7.4 Kompilacja systemu	11
7.5 Konfiguracja i kompilacja aplikacji	12
7.5 Programowanie bitów „fuses” procesora	12
7.6 Programowanie mikroprocesora	13
8. Montaż mechaniczny	13
9. Obsługa kamery	15