

074

A

## ZESPÓŁ AUTOMATYKI ELEKTRONICZNEJ

Nazwa ONB/ZNB

Główny wykonawca

mgr inż. Tadeusz Goszczyński

Wykonawcy:

„Rozwój prac badawczych dotyczących systemu LonWorks”

**Etap 3 . Działania prowadzące do ustalenia partnerów z przemysłu i obiektów , które mogłyby podlegać opracowaniu jako elementy o standardach interfejsu LonWorks w ramach dalszej dwustronnej współpracy z PIAP. Uruchomienie stanowiska Node Bilder do programowania węzłów sieci LonWorks.**

## DOKUMENT WZORCOWY

(Tytuł pracy, numer i tytuł etapu)

KBN

Zleceniodawca

Kierownik Pracowni

mgr inż. Tadeusz Goszczyński

Z-ca Dyrektora  
d/s Bad-Rozwojowych

dr inż. Jan Jabłkowski

Kierownik Zespołu

doc.dr inż. J. Korytkowski

Pracę zakończono dnia 31.10.1998r.

Nr arch. 7592

Nr zlecenia S1834

Analiza deskryptorowa

URZADZENIA AUTOMATYCZNEJ REGULACJI I STEROWANIA +  
SIECIOWE SYSTEMY KOMUNIKACYJNE

Abstrakt

Opis działań marketingowych  
i opis uruchomienia stanowiska Node Bilder

Tytuły poprzednich sprawozdań

„Rozwój prac badawczych dotyczących systemu LonWorks”  
Etap 2. Opracowanie materiałów szkoleniowych w zakresie systemu  
LonWorks umożliwiającym przeprowadzenie szkoleń specjalistycznych  
inżynierów z przemysłu. Nr arch. 7554

Rozdzielnik

Egz. 1. .... OIN  
Egz. 2. .... ZAE - 3  
Egz. 3. .... ZAE - 3  
Egz. 4. .... ZAE - 1

## **Spis treści:**

1. Opis działań marketingowych.
2. Uruchomienie stanowiska NodeBuilder do programowania węzłów sieci LonWorks.
3. Analiza i podsumowanie działań.
4. Uruchomienie interfejsu procesora 8051 do sieci LonWorks.

### Załącznik.

Projekt: „Wykonanie interfejsu szeregowego do komunikacji między procesorem 8051 i Neuron 3150 w celu połączenia centrali typu SC4032 z klawiaturą produkcji firmy SAS poprzez sieć LonWorks.”.

## 1. Opis działań marketingowych.

### 1.1. Kontakty telefoniczne z wybranymi potencjalnymi klientami - użytkownikami systemu SCANLAB-W:

- Laboratorium Instytutu Inżynierii Materiałów Włókienniczych - Łódź. tel.. 042 - 365633  
Kierownik Laboratorium mgr inż. Jadwiga Makowska.
- Laboratorium Badań Właściwości Pożarowych - Józefów. tel.. 022-7893116 w 215  
Kierownik Laboratorium kpt. inż. Małgorzata Suchecka.
- Ośrodek Materiałoznawstwa Korozji i Ochrony Środowiska w Centrum Techniki Okrętowej - Gdańsk Kierownik Ośrodka dr inż. Genowefa Szydłowska-Herbut.
- Laboratorium Wojewódzkiego Inspektoratu Ochrony Środowiska - Sieradz tel.. 043-220981 Kierownik Laboratorium mgr inż. Izabella Racięcka.
- Laboratorium Badawcze Zakładu Technologii i Mechanizacji Przetwórstwa.- Morski Instytut Rybacki. tel.. 058-201728 Kierownik Laboratorium prof. inż. Piotr Bykowski.

Rozmówcy potwierdzali potrzebę zakupu systemu, zgłaszali pytania i potrzeby dotyczące systemu, lecz stwierdzają brak w tym roku środków finansowych na zakup.

### 1.2. Udział w „ IV Sympozjum Klubu Polskich Laboratoriów Badawczych - POLLAB” - Cetniewo, 13-15 maja 1998r. z umieszczeniem opisu systemu monitorowania w materiałach konferencji i prezentacją oprogramowania demonstracyjnego SCANLAB-W dla zainteresowanych uczestników.

Ponadto w celu promocji opublikowano w PAR 4 odcinki artykułu „Sieci mikroprocesorowe od podstaw” autorstwa mgr inż. Tadeusza Goszczyńskiego omawiające sieć LonWorks.

### 1.3. Rozmowy z przedstawicielami firm polskich z branży automatyki i pomiarów dotyczące współpracy przy opracowaniu interfejsu LonWorks do produkowanych przez nich urządzeń:

- SAS- Electronics - Warszawa Właściciel Stefan Zaremba tel.. 6669422.
- Solidex S.A. Kraków - Dyr. Rafał Maciejewicz. tel.. 048- 12368588
- IASE Wrocław Gł. Specjalista dr inż. Andrzej Tomczyk tel.. 071 482183.
- ISS - Katowice z-ca dyr. dr inż. Włodzimierz Boroń tel.. 048-32 581 855
- Nord Sp.z o.o. Gdańsk projektant Marcin Świerczyński tel.. 058-431214
- Akademia Marynarki Wojennej - Gdynia mgr inż. Roman Pałubicki tel.. 058-6262891 i kmdr ppor. mgr inż. Ryszard Studański tel.. 262423.
- KFAP - Kraków Szef Produkcji Zbigniew Szramel tel.. 012- 637 49 07.
- Skamer - Toruń Z-ca Dyrektora mgr inż. Andrzej Turak tel... 014- 212220.
- Skamer - Kraków mgr inż. Ireneusz Stachowicz tel... 012- 2671440.
- Micromex - Konin mgr inż. Andrzej Argasiński tel... 063-454353.
- APS- Białystok prezes zarządu mgr inż. Bogusław Łącki tel... 085-549757.
- OBRME Zielona Góra mgr inż. Stanisław Szymkowiak tel... 068-255258.

Nr arch.. 7592

- Lumel - Zielona Góra Dyrektor Jerzy Woyciechowski tel... 068-253931.
- EMAG Centrum Automatykacji Górnictwa - Katowice mgr inż. Krzysztof Oset tel... 032-1042391.
- Pniefal S.A. - Warszawa Dyrektor Zakładu Elementów Automatyki inż. Antoni Kleniewski fax.: 872-41-94

W efekcie nawiązano współpracę z 2ma firmami:

- SAS - Electronics wycofała się ze współpracy z powodu kłopotów finansowych
- Mera-Pniefal S.A. - zgłosiła chęć współpracy. Trwają uzgodnienia dotyczące sposobu współpracy.

## **2. Uruchomienie stanowiska NodeBuilder do programowania węzłów sieci LonWorks.**

### 2.1 Zestaw.

W skład zestawu NodeBuilder wchodzi:

- Oprogramowanie na dyskietkach.
- Karta do komputera PC.
- Moduł NodeBuilder- LTM-10.
- Zasilacz.
- Pakiet uruchomieniowy Gizmo-3 do testów oprogramowania.
- Komplet kabli do połączeń sieci LonWorks.

NodeBuilder jest przeznaczony dla użytkowników, którzy są zainteresowani opracowaniem urządzeń w systemie LonWorks. NodeBuilder nie zawiera urządzeń do uruchomienia na poziomie systemu takich jak stanowisko LonBuilder ale zawiera kompletny zestaw narzędzi zoptymalizowanych dla opracowania indywidualnych urządzeń pracujących w systemie LonWorks.

### 2.2. Pisanie programu aplikacyjnego w Neuron C.

Język Neuron C jest oparty na ANSI C z rozszerzeniami dla stosowania komunikacji sieciowej i interfejsów do obwodów wejściowo-wyjściowych. Zestaw NodeBuilder zawiera narzędzie zwane Wizard ułatwiające pisanie programów. Wizard wyświetla na ekranie komputera serię formularzy w celu zdefiniowania nowego urządzenia.

Formularz 1 zawiera:

nazwę urządzenia

nazwę katalogu dla programu

Formularz 2 definiuje wzór urządzenia i zawiera:  
dwa gotowe wzory  
opcję wzoru użytkownika

Formularz 3 proponuje opcje:  
generowania kodu programu przez WizarDA  
pisanie kodu programu przez użytkownika

Formularz 4 proponuje do wyboru i dołączenia na życzenie obiekty LonMark dla urządzenia i automatycznie dołącza obiekt węzła.

Formularz 5 proponuje do wyboru i dołączenia na życzenie zmienne sieciowe dla wybranego obiektu

Formularz 6 proponuje dołączenie kodu programu dla obsługi obwodów wejściowo-wyjściowych. Opcja ta zakłada zastosowanie w projektowanym urządzeniu modułu LTM-10, zawartego w zestawie uruchomieniowym NodeBuilder i będącego kompletnym węzłem LonWorks za wyjątkiem specyficznych dla aplikacji obwodów wejściowo-wyjściowych. Moduł ten może mieć dołączone dowolne projektowane przez użytkownika obwody wejściowo-wyjściowe. Jest to opcja polecana dla prototypowej konstrukcji, w przeciwnym wypadku pomijamy ten formularz wybierając klawisz „No, don't add I/O declarations”. Dodawanie kodu dla obsługi obwodów wejściowo-wyjściowych. jest proste i wymaga typowo tylko jednej linii programu na obwód.

Formularz 7 prosi o wpisanie ciągu znaków opisujących węzeł dla dokumentacji własnej węzła, która przechowywana w pamięci własnej urządzenia - węzła sieci LonWorks jest udostępniana na życzenie urządzeniu konfiguracyjnemu sieci.

- Po skompletowaniu tych 7 formularzy program Wizard jest przygotowany do wygenerowania definicji projektowanego urządzenia do swojej bazy danych oraz kodu jego programu.
- Definicja urządzenia zawiera odniesienia do nowego wzoru kodu oraz do wybranego wzoru
- urządzenia dla projektowanego urządzenia.
- Kod źródłowy programu wygenerowanego dla projektowanego urządzenia zawiera następujące elementy:
- komentarze opisujące obiekty LonMark zastosowane w urządzeniu i typy zmiennych sieciowych
- ciąg znaków dokumentacji własnej urządzenia wymieniający obiekty LonMark używane w urządzeniu, w zakodowanej formie dla urządzenia konfigurującego sieć. Na przykład `#pragma set_node_sd_string „@0,1.Temperature sensor.”`
- odniesienie do standardowych zbiorów dołączanych przy kompilacji ( `# include` ) wymaganych dla wybranych typów zmiennych sieciowych.

- deklaracje dla wszystkich wymaganych dla obiektu LonMark zmiennych sieciowych. Te deklaracje zawierają samodefinicje wcześniej opisane dla identyfikacji obiektów LonMark, z którymi są związane a także którym SNVT odpowiadają w obiekcie. Na przykład: network output sd\_string („@1|1.”) SNVT\_temp nvo01Value;
- deklaracje obwodów I/O, jeśli była wcześniej wybrana opcja generowania deklaracji dla modułu LTM-10
- puste zadanie restartu (reset task) - wykonywane za każdym razem gdy urządzenie jest resetowane.
- zadanie zgłoszenia (wink task) - wykonywane na żądanie urządzenia konfigurującego sieć.

Jeśli była wcześniej wybrana opcja generowania deklaracji dla modułu LTM-10 to zostanie wygenerowany kod do mrugnięcia dla diody LED w tym module.

- zadanie uaktualnienia (update task) dla obiektu węzła. Obiekt węzła raportuje status węzła i obiektu przy użyciu mechanizmu żądanie-odpowiedź. Wejściowa zmienna sieciowa jest używana do odbioru żądania statusu a wyjściowa zmienna sieciowa do raportowania statusu. Wytworzony przez program Wizard kod programu urządzenia dekoduje żądanie a zadaniem nowo generowanego kodu jest wytworzenie odpowiedzi właściwej dla aktualnego statusu węzła i obiektu.
- zadanie uaktualnienia zmiennej sieciowej (nv update task) dla każdego obiektu wykonawczego. Te zadania są używane do sterowania urządzeniem wykonawczym
- zadanie zmiany wejścia I/O (I/O change task) dla wyłącznika w module LTM-10, jeśli była wcześniej wybrana opcja generowania deklaracji dla modułu LTM-10
- komentarze „TODO” („Do Zrobienia”) w tych miejscach wygenerowanego kodu, w których powinny być dodane linijki kodu w celu realizacji funkcji aplikacji. Aby urządzenie mogło współpracować z innymi urządzeniami LonWorks pod uwagę należy wziąć następujące możliwości uzupełnień kodu:

opcjonalne deklaracje zmiennych sieciowych. Każdy obiekt ma opcjonalnie zmienne sieciowe, które mogą być zastosowane oprócz wymaganych i generowanych przez program Wizard

deklaracje właściwości konfiguracyjnych. Deklaracje te nie są obowiązkowe, lecz bardzo przydatne dla ustawiania opcji konfiguracji dla każdego obiektu LonMark i dla całego urządzenia. Typowo właściwości konfiguracyjne deklarowane jako zmienne sieciowe o ile ich ilość nie przekracza limitu 62 zmiennych. W takim przypadku właściwości konfiguracyjne są dołączane do zbioru konfiguracyjnego i stosowana jest opcja transferu plików protokołu LonTalk w celu pobrania lub ustawienia właściwości konfiguracyjnych.

deklaracje I/O (wejściowo-wyjściowe).

zadanie reset. Kod inicjalizujący aplikację i ewentualnie obwody I/O. To zadanie powinno być krótkie aby uniknąć błędów przy instalacji urządzenia. W razie konieczności wykonywania długiej sekwencji zadań przy restartowaniu urządzenia sekwencja ta powinna być podzielona na niezależne zadania korzystając z mechanizmu planowania (scheduler) mikroukładu Neuron.

zadanie zgłoszenia się urządzenia (wink task). Zadanie to nie jest obowiązujące ale bardzo przydatne dla urządzeń, które mogą być instalowane w różnych systemach. Powinno dawać fizyczne wskazanie lokalizacji urządzenia dla instalatora systemu sieciowego. Może to być zapalenie i zgaszenie lampki, piśnięcie komputera lub mrugnięcie LED stosowanego jako wskaźnik serwisowy w większości urządzeń LonWorks.

zadanie żądania w obiekcie węzła. Program Wizard generuje wzór kodu programu do obsługi nadchodzących z sieci żądań dla obiektu węzła. Do tego wzoru powinien być dodany kod raportujący informację o statusie i zapewniający kontrolę obiektów w urządzeniu LonWorks.

zadanie wejściowej zmiennej sieciowej. Program Wizard generuje wzór kodu programu do obsługi dla obiektów wykonawczych. Do wzoru powinien być dodany kod się i zapewniający kontrolę nad tymi urządzeniami w zależności od otrzymanych z sieci wartości sterujących.

zadanie kontroli wejściowych obwodów. Obwody wejściowe są typowo wyzwalane przez zdarzenia ( events ) pochodzące od samych obwodów wejściowych, lub przez programowe timery, które wyzwalają zadania tajmerów. Na przykład wejście wyłącznika może aktywizować zadania I/O. Jednakże, zadanie timera może być użyte w celu okresowego sprawdzania czujnika temperatury lub innego wejścia analogowego.

### 2.3. Budowanie, Testowanie, Wykrywanie błędów w pojedynczych urządzeniach.

Po wykonaniu czynności opisanych w poprzednich punktach można wykonać prototyp urządzenia LonWorks korzystając z modułu LTM-10 a następnie uruchomić na nim program aplikacyjny.

Wybranie z menu głównego pozycji Build spowoduje kompilację, linkowanie z odpowiednimi bibliotekami, budowanie wersji do załadowania do pamięci urządzenia tzw. memory image, instalowanie urządzenia jako węzła sieci LonWorks, tak aby mogło się komunikować z komputerem PC, na którym pracuje program NodeBuilder i wreszcie ładuje „memory image” do urządzenia. Jedna komenda wywołuje wszystkie te czynności z załadowaniem programu aplikacyjnego do urządzenia i przygotowuje je do testowania. Urządzenie LonWorks testowane jest wykorzystując narzędzie NodeBuilder'a stosujące przeglądarkę zmiennych sieciowych. Przeglądarka ( browser ) przedstawia w formie podobnej do arkuszy Excela wszystkie zmienne sieciowe urządzenia wraz z ich bieżącymi wartościami i umożliwia zmianę tych wartości przy pomocy klawiatury komputera. Przeglądarka wykorzystuje opisane poprzednio mechanizmy programu LonManager DDE Server. Przeglądarka zmiennych sieciowych bada urządzenie LonWorks poprzez jego interfejs zewnętrzny.

Na przykład odpowiedzi nadchodzące do wejściowych zmiennych sieciowych obiektów wykonawczych mogą być weryfikowane przez zmianę zmiennych sieciowych w przeglądarce i bieżące obserwowanie odpowiedzi. Także obiekty czujników mogą być testowane poprzez obserwowanie wartości zmiennych sieciowych na przeglądarce i porównywanie ich do fizycznych wartości.



Jeśli przeglądarka wykazuje, że urządzenie nie pracuje właściwie, NodeBuilder Neuron C Debugger powinien być wykorzystany w celu wykrycia błędów. Program Debugger zapewnia przegląd kodu źródłowego programu aplikacyjnego, zastawianie pułapek, startuje i zatrzymuje program, uruchamia program aplikacyjny krok po kroku i przedstawia na ekranie komputera widok modyfikowanych obiektów takich jak zmienne sieciowe i timery.

#### 2.4. Integrowanie sieci urządzeń LonWorks i testowanie sieci.

Kiedy już pojedyncze urządzenia LonWorks zostaną uruchomione i przetestowane można je integrować w większą sieć LonWorks. Urządzenia LonWorks nie mają zafiksowanych hardwarowo adresów. Urządzenie konfiguruje sieć LonWorks przydziela adresy sieciowe obiektom LonMark w czasie ich instalacji bez wpływania na wykorzystywaną aplikację w urządzeniu LonWorks. Typowo stosuje się w tym celu narzędzia zainstalowane na komputerze PC takie jak np. LonMaker Installation Tool, które instaluje w sieci wszystkie urządzenia LonWorks i ustala pomiędzy nimi połączenia logiczne. Następnym krokiem w uruchomieniu sieci może być zastosowanie narzędzia diagnostycznego sieci LonWorks takiego jak LonManager Protocol Analyzer, które zainstalowane na komputerze PC umożliwia monitorowanie przepływu przesyłek w sieci LonWorks i weryfikowanie poprawności pracy tej sieci.

### 3. Analiza i podsumowanie działań.

#### 3.1. SCANLAB - System monitorowania laboratoriów.

Jak wykazały prace Agencji Innowacji Marketingowych „Informacje i Strategie” przeprowadzone w grudniu 1997 roku wykonywane na zlecenie PIAP na 103 przebadane laboratoria tylko 3 wykorzystywały systemu monitorowania warunków środowiskowych. Badania te wskazały na 8 laboratoriów, które w ankiecie wskazywały na zainteresowanie zakupem lecz Agencja uważała sprzedaż 3-4 kompletów tylko w bardzo optymistycznym scenariuszu. Wyniki naszych dalszych kontaktów potwierdziły zainteresowanie i optymistyczne nastawienie kierowników laboratoriów, którzy nadal przewidują zakup systemu w przypadku korzystnego finansowego wyniku przedsiębiorstwa w br. Ocena rynku polskiego w dziedzinie monitoringu wykonana przez firmę Informacje i Strategie wykazuje, że w całej Polsce w ciągu roku sprzedano 46 systemów tego typu w elektrociepłowniach, stacjach redukcji gazu, kotłowniach i zakładach przemysłu spożywczego. Systemy te zostały sprzedane przez prywatne firmy znajdujące się od wielu lat na tym rynku i mające do przedstawienia klientom wiele pracujących systemów jako referencje. Doświadczenia ZAE, potwierdzone przez OAP wskazują, że bez bardzo silnej akcji promocyjnej wejście na nowe rynki jest w tej branży bardzo trudne.

#### 3.2. Kontakty z kandydatami do konsorcjum OSA.

Niestety z nieznanych powodów - każdy z kandydatów obarcza winą Agencję i innych partnerów - nie ma żadnych pozytywnych skutków działań Agencji. Wg. naszej wiedzy nie powstał zapowiadany business plan ani nie zawiązało się konsorcjum. W tej sytuacji argumenty potencjalnych naszych klientów są stale te same: co pozytywnego wyniknie dla nas z wprowadzenia interfejsu LonWorks do naszych produktów, jeśli nasz produkt będzie jedyny na polskim rynku z tym interfejsem to nic na tym nie skorzystamy.

Niestety nie ma nacisku projektantów systemów na producentów aparatury w tej sprawie. Wykorzystują to firmy polskie o zagranicznym pochodzeniu, takie jak TAC ściśle współpracująca z macierzystą Tour Anderson w Szwecji i instalują systemy oparte na zagranicznej aparaturze wykorzystującej technologię LonWorks. Także Honeywell i ABB instalują systemy wykorzystujące tę technologię. Jednocześnie firma Scamer z Tarnowa, bardzo aktywna w tej dziedzinie współpracuje z bliską sobie geograficznie ( Scamer ma oddział w Krakowie ) AGH szkoląc młodych pracowników w projektowaniu nowych urządzeń w technologii LonWorks. Potwierdza ona nadal zainteresowanie współpracą z PIAP lecz bez angażowania środków finansowych. Jedyne kontakty z pobliską MERA-PNEFAL wydaje się rokować nadzieje. Naszym zdaniem należy postępować podobnie jak Scamer i AGH. Przeprowadzenie przez nas samego szkolenia pracowników PNEFAL może zakończyć się podjęciem przez nich samodzielnego opracowania systemu opartego na tej technologii, gdyż czasu na naukę mają dużo - ich systemy sprzedają się jak dotąd bez interfejsu Lon zadawalająco. Natomiast przy umowie o współpracy i zatrudnieniu w ramach tej pracy młodego pracownika (absolwenta lub studenta IV roku) w PIAP ZAE mogłoby rozpocząć wieloletnią współpracę. Trzeba tu zwrócić uwagę na konkurencyjne rozwiązanie jakie wg. naszych informacji wybrała firma LabEl - Piastów . Zamówiła ona w ramach umowy o dzieło ze studentem P.W. opracowanie interfejsu LonWorks do swoich przetworników. Nie wiadomo jeszcze jakie będą następstwa tego kroku.

### 3.3. Próba wdrożenia technologii LonWorks do systemu alarmowego.

Firma SAS-electronics, której właściciel bardzo aktywnie działał na rzecz unowocześnienia swojego sprzedawanego od kilku lat systemu, niestety została przez sytuację rynkową w tej branży zmuszona do wycofania się z przygotowanej już umowy. Koszt unowocześnienia związany ze zmianą oprogramowania i sprzętu w systemie, w celu zapewnienia mu konkurencyjnych parametrów technicznych ( niezależnie od dalszych kosztów wprowadzenia transmisji w technologii sieci LonWorks ) okazał się zbyt wysoki. Jak nas poinformowano osiągalne na polskim rynku amerykańskie systemy alarmowe są tańsze od odpowiadających im poziomem technicznym nowym systemom opracowywanym w kraju.

### 3.4. Propozycje dalszych działań.

Uważamy, że współpraca z partnerem z przemysłu powinna się odbywać przez pewien czas przy minimalizacji wymaganych od partnera nakładów finansowych, tak by nie wybrać rozwiązań konkurencyjnych. Jednocześnie z tym powinien powstać 3 osobowy zespół (zawierający także młodych ludzi) w PIAP (lub grupa międzyzakładowa) , który gwarantowałby wykonanie podjętych przyszłych zobowiązań niezależnie od urlopu lub choroby jednego pracownika.

Może dopiero uzyskany projekt Celowy z KBN lub Unii Europejskiej pozwoli na realizację takiego scenariusza, dlatego rozpoczynamy intensywne działania nawiązywania w tym celu kontaktów z firmami zagranicznymi. Dla wsparcia tych działań korzystne byłoby naszym zdaniem:

- zorganizowanie stale działającego „klubu wnioskodawców” ze stałym konsultantem - dla ciągłej wymiany doświadczeń i tworzenia projektów interdyscyplinarnych,
- utworzeniu anglojęzycznej strony WWW dla PIAP i grup tematycznych,
- zgłoszenie PIAP do bazy Partnerów w CORDIS i wspomaganie zgłoszeń tematycznych do tej bazy.

#### **4. Uruchomienie interfejsu procesora 8051 do sieci LonWorks.**

Opis pracy został umieszczony w Załączniku. 1.

Nr arch. 7592

Maciej Bornikowski  
Grzegorz Wójtowicz

M. Bornikowski  
G. Wójtowicz

Warszawa 1998. 07. 30

### Projekt

Wykonanie interfejsu szeregowego do komunikacji między procesorem 8051 i Neuronem 3150 w celu połączenia centrali typu SC 4032 z klawiaturą produkcji firmy SAS poprzez sieć LON.

Celem projektu było wykonanie prototypowego zestawu klawiatura - centrala systemu alarmowego firmy SAS, w którym transmisję szeregową między nimi zastąpić miała transmisja po sieci LON z wykorzystaniem standardowych modułów LTM-10 z transceiverami TP/XF 78. Centrala i klawiatura w pierwotnej wersji komunikowały się na zasadzie odpytywania, przesyłając w postaci szeregowej ciągi znaków w formacie EIA-232. Bez ingerencji w programy działania centrali i klawiatury wprowadzono do komunikacji dwa moduły z procesorami Neuron, które w z punktu widzenia centrali i klawiatury miały stworzyć przezroczysty układ transmisji danych. Założenie było takie, że Neuron miał odbierać kolejne znaki wysyłane po RS-ie i w postaci zmiennych sieciowych transmitować je do drugiego Neurona, gdzie z kolei zamieniane były na postać szeregową. System miał być dwukierunkowy. Został on wykonany, oprogramowany i skonfigurowany. Centrala z klawiaturą komunikowały się, jednak transmisja ta była niedoskonała, gdyż podczas komunikacji gubione były niektóre znaki. Stan taki był rezultatem sposobu w jaki zorganizowana jest transmisja szeregową po stronie procesora Neuron. Nie można w przypadku tych procesorów korzystać z mechanizmu przerw, gdyż nie ma takiego mechanizmu. Można jedynie w każdym obiegu sprawdzania kolejnych zdarzeń przez „scheduler” odczytywać z wejścia szeregowego zadaną liczbę znaków, przy czym jeżeli żaden znak się nie pojawia funkcja oczekuje na jego przyjscie przez okres równy odbiorowi 20 znaków i kończy swe działanie. Jeśli więc program zajęty jest wykonywaniem innego fragmentu programu znak zostanie zgubiony. Były podejmowane próby wykrywania nadchodzącego znaku i odczytywania portu tylko w takich przypadkach. Rozwiązanie takie usprawniło nieco transmisję, ale nie wyeliminowało całkowicie jej wad. Dla zapewnienia właściwej transmisji dla takiego przypadku konieczne byłoby zastosowanie dodatkowego modułu PSG-10 lub PSG/2, który umożliwiłby buforowanie do 16 znaków i zwiększenie szybkości transmisji do 115,2 kbps. Rozwiązanie takie wydaje się jednak nieopłacalne. Innym

sposobem jest zmiana protokołu wymiany danych między procesorami, który w powyższym przypadku przystosowany był do konkretnych zadań, nie nadawał się natomiast do komunikacji z Neuronem.

Nasuują się więc trzy rozwiązania komunikacji między procesorem z rodziny 8051 i Neuronem :

- Pierwszym z nich jest komunikacja szeregową jak w powyższym przypadku, z wprowadzeniem dodatkowego „hanshakingu” za pomocą dodatkowych połączeń (2 linie sygnałowe).
- Drugim jest transmisja równoległa poprzez jeden z portów procesora (8 bitowa szyna danych) i dodatkowych 2 linii sterujących.
- Trzecim jest umieszczenie procesora Neuron w obszarze zewnętrznej pamięci danych, przy czym komunikacja odbywałaby się poprzez zapis lub odczyt komórki pamięci. Takie rozwiązanie wymagałoby również wprowadzenie 3 linii sterujących.

Dwa pierwsze rozwiązania nie wymagają stosowania dodatkowych układów i są dobrym rozwiązaniem w przypadku stosowania procesorów z wewnętrzną pamięcią programu. W przypadku stosowania zewnętrznej pamięci programu i ograniczonej ilości wejść wyjść najlepszym jest rozwiązanie trzecie, ale wymusza ono zastosowanie dodatkowego układu zatrzymującego z wyjściami trójstanowymi dla zapewnienia prawidłowego odczytu programu. Rozwiązanie pierwsze w przypadku wykorzystywania przez system wejścia-wyjścia szeregowego jest również utrudnione i wymusza w takim przypadku multipleksowanie tego wyjścia-wejścia.

Spośród proponowanych rozwiązań została wykonana i przetestowana komunikacja równoległą przez port P1(80C51) i IO10-7(NEURON3150) oraz szeregową przez port szeregowy RxTx(80C51) i IO8,10(NEURON3150) jako wejścia/wyjścia szeregowo.

Proponowane rozwiązania dołączone są w postaci schematów.

Listingi programów dla 80C51 i dwóch procesorów NEURON3150 załączono do dokumentacji.

```

//***** Obsluga komunikacji 80C51 <-> NEURON 3150
//***** Program dla mikrokontrolera 80C51
//***** ver.1.0 autor: Maciej Bornikowski
//***** 08.98 Grzegorz Wojtowicz

#pragma SYMBOLS
#include <reg51.h>
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

bit tim1;
bit bmtim;
data unsigned char czas;
data unsigned char ms=40;
data unsigned char mtim=50;
data char c;
data char dana_we2;

bit flaga_RD;
bit flaga_RS;

sbit WRbit = P3^4;

//*****Inicjalizacja mikrokontrolera*****
void init()
{
    SCON = 0x50;
    PCON = 0x80;
    TMOD = 0x22;
    TCON = 0x51;
    IE = 0x93;
    IP = 0x03;
    TH0 = 5;
    TH1 = 243;
    TR0 = 1;
    TR1 = 1;
    TI = 1;
}

//*****Obsluga przerwania od timera*****
void timer0(void) interrupt 1
{
    if(tim1==1)
    {
        ms--;
        if(ms==0)
        {
            ms=40;
            mtim--;
            if(mtim==0)
            {
                mtim=10;
                bmtim=1;
            }
            if(czas!=0) czas--;
        }
    }
}

//*****Obsluga przerwania zewnetrznego od NEURONA*****
void zmianaRD(void) interrupt 0
{

```

```

        flaga_RD=1;
        P1=0xFF;
        dana_we2=P1; //Odczytanie znaku z portu rownoleglego
    }

//*****Funkcja opozniajaca*****
void pause(unsigned char t)
{
    timl=1;
    czas=t;
    while(czas);
    timl=0;
}

//*****Glowny program*****
void main()
{
    init(); //Inicjalizacja mikrokontrolera
    printf("\n\n\n          Program do sprawdzenia komunikacji miedzy\n")
;
    printf("          procesorem 8051, a Neuronem.\n\n");
    flaga_RD=0;
    while(1)
    {
        if(RI==1)
        {
            if(flaga_RD==0)
            {
                WRbit=0;
                c=SBUF; //Odebranie znaku z portu szeregoweg
o
                putchar(c);
                pause(1);
                P1=c; //Wyslanie znaku do NEURONA
                WRbit=1;
                pause(1);
                RI=0;
            }
        }
        //Wyslanie odebranego od NEURONA znaku na port rownolegly
        if(flaga_RD==1)
        {
            putchar(dana_we2);
            flaga_RD=0;
        }
    }
}

```

```
//Program dla procesora Neuron komunikujacego sie poprzez port
//rownolegly z procesorem 8051
```

```
IO_0 output byte dana_wy;
IO_0 input byte dana_we;

IO_8 input bit WR;
IO_10 output bit RDint;
network input char dana_in;
network output char dana_out;
network output boolean odczyt_out;
network output unsigned long licznik_out;
boolean odczyt;

void wyslij_dane(void)
{
    io_set_direction(dana_wy, IO_DIR_OUT);
    io_out(dana_wy, dana_in);
    io_out(RDint, 0);
    io_out(RDint, 1);
    io_set_direction(dana_wy, IO_DIR_IN);
}

void odbierz_dane(void)
{
    io_set_direction(dana_we, IO_DIR_IN);
    dana_out=io_in(dana_we);
    odczyt_out=odczyt;
    odczyt=!odczyt;
}

when(reset)
{
    odczyt=TRUE;
    io_out(RDint, 1);
    licznik_out=0;
}

when(nv_update_occurs(dana_in))
{
    if(io_in(WR)!=0)
    {
        wyslij_dane();
    }
}

when(io_changes(WR) to 1)
{
    odbierz_dane();
    licznik_out++;
}
```



```
//Program dla procesora Neuron przesyłającego informacje do drugiego  
//Neurona współpracującego z 8051
```

```
#include<control.h>
```

```
network input char dana_in;  
network output char dana_out;  
network input unsigned long rtime_in;  
mtimer repeating tim=5000;  
unsigned int dana;  
boolean led_act;
```

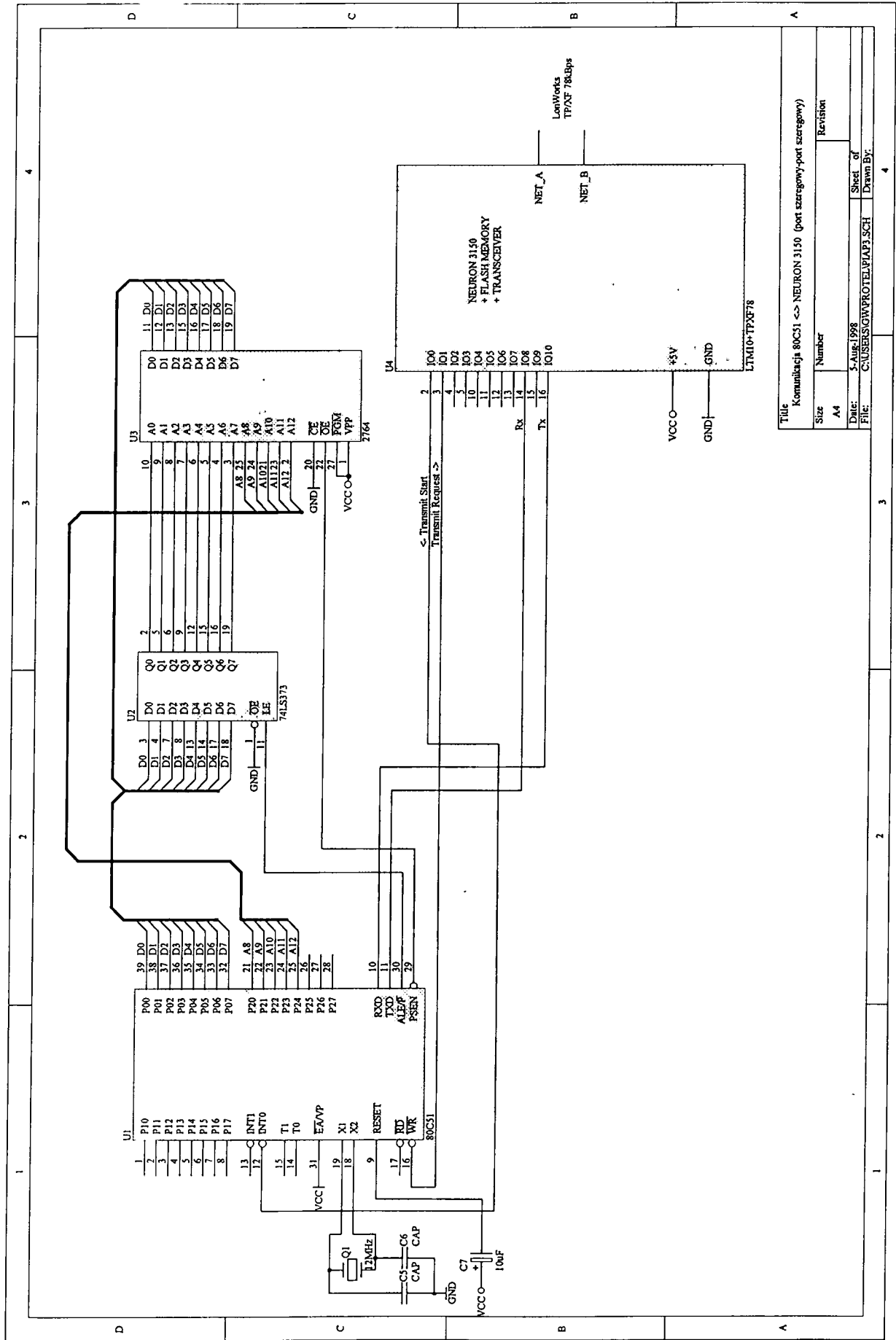
```
when(reset)  
{  
    dana=97;  
    led_act=1;  
}
```

```
when(timer_expires(tim))  
{  
    dana++;  
    if(dana>122) dana=97;  
    dana_out=dana;  
}
```

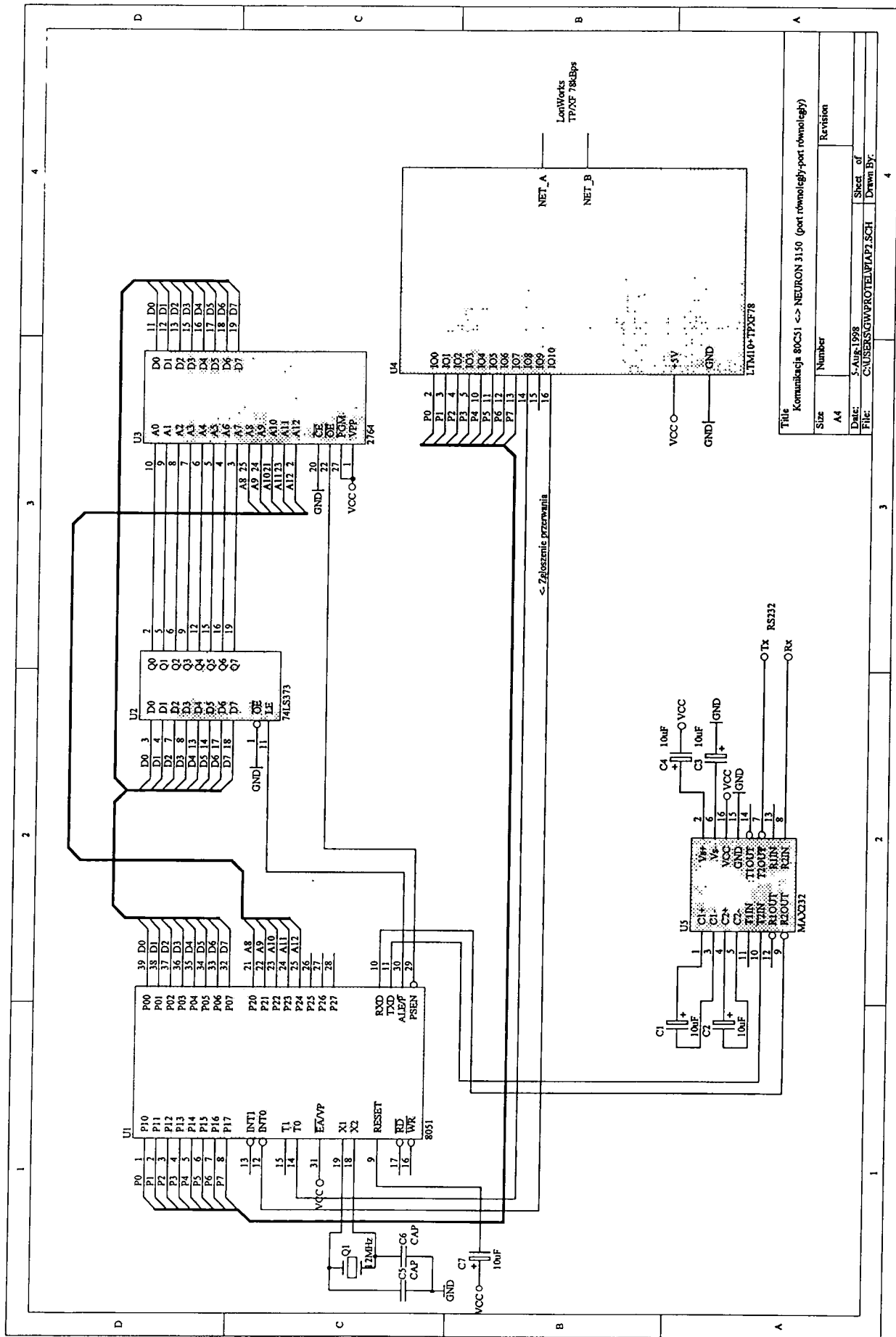
```
when(nv_update_occurs(dana_in))  
{  
    dana_out=0;  
    if(dana_in==1)  
    {  
        activate_service_led=led_act;  
        led_act=!led_act;  
    }  
}
```

```
when(nv_update_occurs(rtime_in))  
{  
    tim=rtime_in;  
}
```

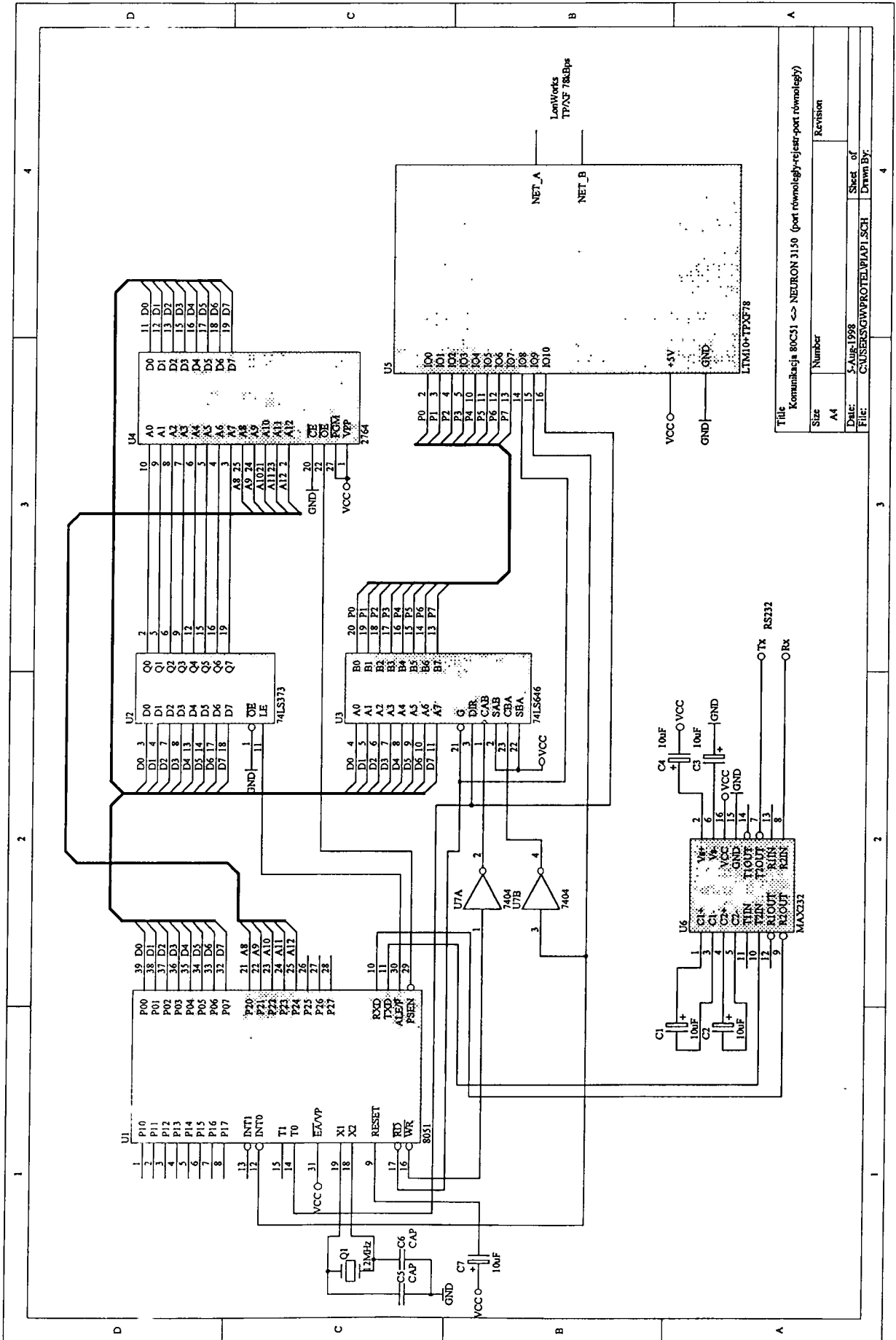
```
□
```



Title		Komunikacja 80C51 <-> NEURON 3150 (port szeregowy-port szeregowy)	
Size	Number	Revision	
A4			
Date:	5-Aug-1998	Sheet of	
File:	C:\USER\G\PROTEL\PIAP3.SCH	Drawn By:	



Title		Komunikacja 80C51 <-> NEURON 3150 (port równoległy-port równoległy)	
Size	Number	Revision	
A4			
Date:	Sheet of	Drawn By:	
5-Aut-1998	1	C:\USER\SGW\PROTEL\AP2\SCH	



Title		Komunikacija 80CS1 ↔ NEURON 3150 (port rđwnologjy-tejstr-port rđwnologjy)	
Size	Number	Revision	
A4			
Date:	5-Aug-1998	Sheet of	4
File:	C:\SEBS\GWP\PROTEL\PIPI\SCH	Drawn By:	