

ZESPÓŁ LABORATORIÓW TELEINFORMATYKI TRANSPORTU

**ZAKŁAD INŻYNIERII TRANSPORTU LOTNICZEGO
I TELEINFORMATYKI (ITLIT)**

**Politechnika
Warszawska**

**Wydział
Transportu**



LABORATORIUM ZITLIT

INSTRUKCJA DO ĆWICZENIA NR 15

Systemy transponderów zbliżeniowych RFID

© ZITLIT WT PW, DO UŻYTKU WEWNĘTRZNEGO

Warszawa 2023

1. Cel i zakres ćwiczenia

Celem ćwiczenia jest zapoznanie z podstawowym układem systemów transponderów zbliżeniowych, jakim jest np. układ czytnik-karta zbliżeniowa, masowo wykorzystywanych w rozwiązaniach z dziedziny identyfikacji, kontroli dostępu, elektronicznej płatności. Podczas ćwiczenia prezentowane są następujące zagadnienia:

- transponder – np. karta zbliżeniowa,
- czytnik transponderów - kart zbliżeniowych,
- format i protokół transmisji danych,
- odbiór danych (kodu) z transpondera,
- możliwości wykorzystania,
- przykłady oprogramowania.

2. Wykaz wykorzystanych przyrządów

- komputer PC z systemem Windows 10,
- komputerowa przystawka oscyloskopowa PCSU1000 (rys. 2.1)
- multimetr elektroniczny (rys. 2.2),
- czytnik transponderów - kart zbliżeniowych HB2300 (rys. 2.3),
- zestaw kart (i innych elementów) zbliżeniowych (rys. 2.4),
- oprogramowanie transmisji szeregowej ComTest (rys. 2.5).



Rys. 2.1 Komputerowa przystawka oscyloskopowa PSC500



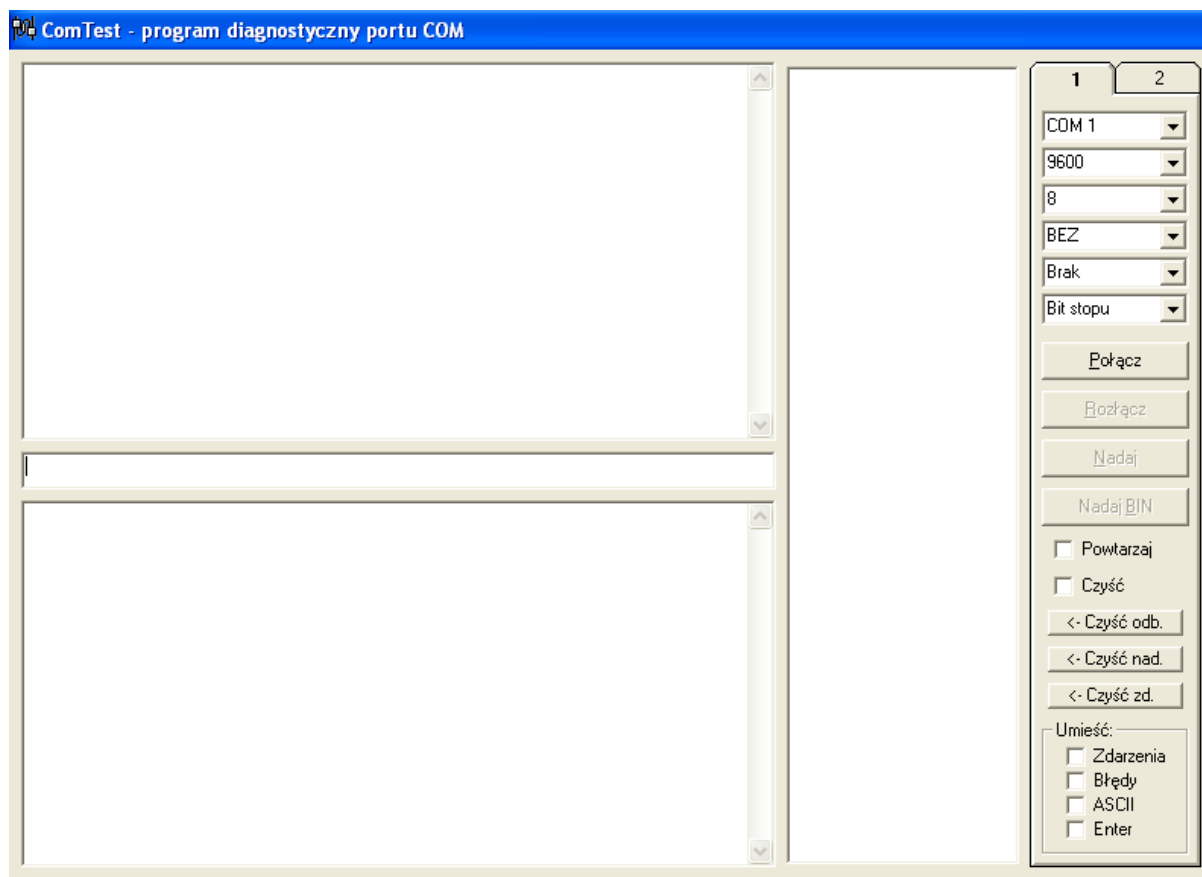
Rys.2.1 Multimetr elektroniczny



Rys. 2.2 Czytnik transponderów HB2300 z modulem odbiorczym TRD-80 Classic



Rys. 2.3 Zestaw transponderów/kart (i innych elementów) zbliżeniowych



Rys. 2.4 Oprogramowanie transmisji szeregowej ComTest

3. Wprowadzenie

3.1 Łącze RS232C – podstawy teoretyczne (wg <http://www.arczer.neostrada.pl/komputery.htm> oraz <http://forbot.pl/blog>)

Historia

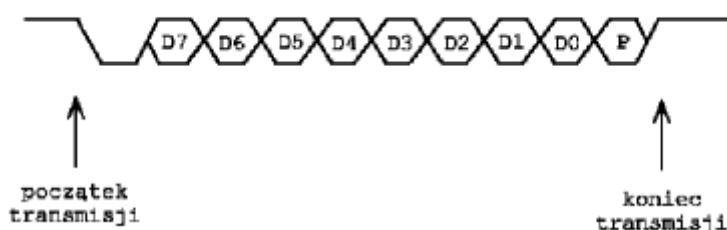
Standard RS-232 (Recommended Standard) został wprowadzony w 1962 roku. Został on opracowany na potrzeby obsługi modemów (o czym świadczą nazwy sygnałów sterujących) przez amerykańskie stowarzyszenie EIA (Electronic Industries Association). W sierpniu 1969 roku wprowadzono zrewidowaną normę oznaczoną RS-232C, która reprezentuje powszechnie akceptowany sposób transmisji danych na nieduże odległości (do 15m), z szybkością do 115 kbitów/s. Standard RS-232C został później przyjęty w Europie przez komitet normalizacyjny CCITT i otrzymał oznaczenie V24. W komputerach osobistych stał się on standardem łącza szeregowego. W systemach operacyjnych portom takim przyznano nazwy logiczne COMn (gdzie n oznacza numer portu). Oprócz obsługi modemów, interfejs umożliwia podłączenie takich urządzeń jak mysz czy drukarka. Może posłużyć

również do połączenia bezpośrednio dwóch komputerów (przez tzw. Kabel Null-Modem)

Sposób transmisji asynchroniczny

- Przy transmisji asynchronicznej urządzenie odbiorcze synchronizuje się podczas odbioru każdego znaku
- Synchronizację uzyskuje się przez dodanie do każdego przesyłanego znaku bitu startu 0 oraz 1, 1.5, 2 bitów stopu
- Kontrolę poprawności zapewnia bit parzystości; dostępne są następujące sposoby wyznaczania wartości bitu parzystości:
 - suma bitów włącznie z bitem parzystości jest nieparzysta (Parity Odd),
 - bit parzystości zawsze równy 0 (Parity Space),
 - bit parzystości zawsze równy 1 (Parity Mark),
 - brak bitu parzystości – nie jest umieszczany w ramce (Parity None).

Pojedyncza paczka informacji



- Czas trwania pojedynczego bitu nazywany jest odstępem jednostkowym. Jego odwrotność określa szybkość transmisji w bodach (bitach na sekundę). Typowe prędkości transmisji wynoszą: 300, 1200, 2400, 9600, 14400, 28800, 56600,
- Długość pola danych może wynosić od 5 do 8 bitów,
- Transmisje kończą jeden lub dwa bity stopu.

Sposób transmisji synchroniczny

- przy transmisji synchronicznej znaki są przesyłane blokami,
- gdy nie ma znaków do wysłania transmitowana jest sekwencja synchronizująca składająca się z jednego lub dwóch znaków synchronizacji, tzw. puste znaki,
- te znaki są ignorowane przez urządzenie odbierające,
- możliwa jest płynna regulacja prędkości transmisji.

Parametry elektryczne

Poziomy logiczne interfejsu różnią się od poziomów zastosowanych wewnątrz komputera. Zastosowano napięcia wyższe i o różnym znaku. Logiczna '1' to napięcie z zakresu -3 do -15V Logiczne '0' to napięcie od +3 do +15V. Zwiększenie wartości napięć i zastosowanie napięć o obydwu znakach pozwoliło zwiększyć odporność sygnału na zakłócenia i przesyłać informacje na większe odległości.

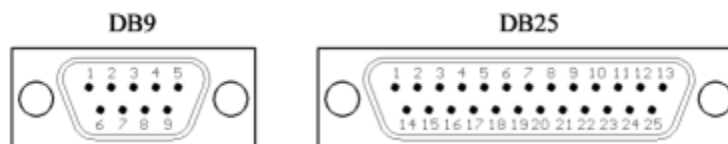
Typy urządzeń

Ponieważ łącze RS232C wywodzi się od modemów, przy jego opisie stosowana jest terminologia związana z transmisją informacji pomiędzy komputerami przy użyciu modemów. Stąd też wyróżnia się dwa rodzaje urządzeń:

- urządzenia DTE (Data Terminal Equipment) będące końcowymi urządzeniami dla przesyłanej informacji (np. komputer),
- urządzenia DCE (Data Communication Equipment) pośredniczące między urządzeniem DTE a siecią telekomunikacyjną (modem).

Realizacja fizyczna (złącza)

W łączu RS232C najczęściej stosowane jest 25-stykowe złącze szufladkowe typu CANNON DB-25P lub DB-25S. Często spotyka się również 9-stykowe złącze szufladkowe typu DB-9, na które wyprowadzono tylko najważniejsze sygnały przeznaczone do asynchronicznej transmisji metodą start-stop.



Topografia wyprowadzeń sygnałów dla łącza RS-232C

Wtyk DB-25	Wtyk DB-9	Nazwa sygnału	Kierunek sygnału
1	-	Masa ochronna	-
2	3	Dane wysyłane (TxD)	wy
3	2	Dane otrzymywane (RxD)	we
4	7	Gotowość wysłania danych (RTS)	wy
5	8	Gotowość przyjęcia danych (CTS)	we
6	6	Gotowy zestaw danych (DSR)	we
7	5	Masa sygnałowa (SG)	-
8	1	Wybrany (RLSD)	we
20	4	Urządzenie gotowe (DTR)	wy
22	9	Sygnał dzwonienia (RI)	we

W magistrali interfejsu RS-232C można wyróżnić kilka grup linii:

- linie danych,
- linie sterujące,
- linie synchronizacji,
- linie masy.

Linie danych

Do dwukierunkowego przesyłania danych przeznaczone są 4 linie. Dwie z nich (TxD, RxD)

tworzą kanał podstawowy, dwie pozostałe (STxD, SRxD) - kanał powrotny, nie wykorzystywany w przypadkach bezpośredniej współpracy komputer-terminal. Funkcje linii kanału podstawowego są następujące:

TxD (2) - *dane nadawane*. Linia wykorzystywana jest do przesyłania danych przez DTE. Standard wymaga, aby w odstępach między przesyłanymi danymi linia była w stanie logicznym "1". Transmisja może odbywać się tylko wtedy, gdy aktywne są sygnały CTS, DSR, DTR i RTS.

RxD (3) - *dane odbierane*:- Linia wykorzystywana jest do przesyłania danych przez DCE. Linia ta powinna być w stanie logicznym "1", gdy linia DCD jest w stanie pasywnym. W przypadku jednej linii transmisyjnej między urządzeniami, po której oba urządzenia mogą przesyłać dane (oczywiście nie równocześnie), linia RxD powinna być w stanie "1", gdy aktywny jest RTS.

Linie sterujące

W tej grupie linii najistotniejsze ze względu na połączenie komputer-terminal są linie przekazujące sygnały gotowości urządzeń do pracy (DSR, DTR) oraz sygnały gotowości do transmisji (RTS, CTS).

DSR (6) - *gotowość DCE*. Stan aktywny oznacza gotowość DCE do współpracy, tzn. gotowość do wymiany dalszych sygnałów sterujących w celu dokonania transmisji danych. Nie oznacza to jednak, że istnieje gotowy zbiór danych, które DCE chce przesłać do DTE. Jest to jedynie informacja o braku przeszkody do transmisji (nie zaistniał żaden defekt).

DTR (20) - *gotowość DTE*. Stan aktywny oznacza gotowość DTE do współpracy z DCE, rozumianej podobnie jak dla linii DSR.

RTS (4) - *żądanie nadawania*. Stan aktywny tej linii oznacza, że DTE zgłasza do DCE żądanie wysłania danych. Powoduje to załączenie przez DCE sygnału CTS. Dane nie mogą być przesyłane, jeśli nie została aktywnie ustawiona linia CTS. Po przejściu sygnału RTS w stan pasywny nie powinien on zostać powtórnie załączony, dopóki DCE nie wycofa aktywnego sygnału CTS.

CTS (5) - *Gotowość do nadawania*. Za pomocą tej linii DCE może zgłaszać do DTE swoją gotowość do odbioru danych z DTE. Przy bezpośredniej współpracy terminal-komputer stan

aktywny tej linii oznacza gotowość komputera do przyjmowania danych, a przy pracy z modemem - gotowość do nadawania przez DCE informacji do odległego urządzenia DCE. Linie RTS i CTS mają swoje odpowiedniki wśród linii kanału powrotnego (SRTS i SCTS), nie wykorzystywanego przy bezpośredniej współpracy komputer-terminal. Na uwagę zasługują również dwie linie związane z poprawnością sygnałów odbieranych: DCD i CG (ozn. zgodnie z RS-232C).

DCD (8) - *poziom sygnału odbieranego*. Linia ta jest wykorzystywana zasadniczo tylko przy współpracy z modemem. Stan aktywny jest generowany przez modem (DCE) i przekazywany do DTE w przypadku odebrania przez modem poprawnego sygnału częstotliwości nośnej z kanału transmisyjnego, co oznacza, że sygnał odbierany z tego kanału przez DCE znajduje się w zakresie wartości prawidłowych. Przy współpracy komputera z terminalem (urządzeniem) może jednak zachodzić konieczność załączenia stanu aktywnego na tej linii. Linia DCD (ozn. również RLSD) ma swój odpowiednik w kanale powrotnym: SRLSD (12).

CG (21) - *jakość sygnału odbieranego*. Linia tą są przekazywane dla DTE informacje o jakości sygnału odbieranego z linii transmisyjnej przez DCE. Stan aktywny linii CG informuje, że transmisja danych prawdopodobnie odbywa się bez błędów. Natomiast stan nieaktywny oznacza, że istnieje duże prawdopodobieństwo przekłamania danych.

Pozostałe linie sterujące nie są w zasadzie wykorzystywane w systemach pomiarowych. Dotyczą one wyboru szybkości transmisji przez DTE lub DCE (ozn. CH/CI zgodnie z RS232C) oraz wskaźnika wywołania RI.

CH/CI (23) - *wybór szybkości transmisji*. Linia CH/CI służy do wyboru prędkości transmisji między dwiema szybkościami, które mogą być dostępne w przypadku transmisji synchronicznej lub dwoma zestawami prędkości, które mogą być dostępne w przypadku transmisji asynchronicznej. Linia ta może być sterowana albo przez DTE albo przez DCE. DCE może określać szybkość transmisji DTE, natomiast DTE - szybkość transmisji (nadawania i odbioru) realizowanej przez DCE. Jeśli źródłem sygnału jest DTE, a odbiorcą DCE, to linia nosi nazwę CH; w przeciwnym przypadku - CI. Stan aktywny sygnału oznacza wybór większej z dwu dostępnych prędkości lub większego z dwu dostępnych zestawów.

RI (22) - *wskaźnik wywołania*. Za pomocą tej linii DCE może informować DTE o tym, czy odbiera sygnał z oddalonego DCE.

Linie synchronizacji

Przesyłanie informacji po liniach danych może być realizowane asynchronicznie lub synchronicznie. Przy transmisji synchronicznej wykorzystuje się linie podstawy czasu, którymi przesyłane są tzw. sygnały zegarowe. .

Na złączu 25-stykowym występują trzy linie tego typu, oznaczone zgodnie z RS232C jako:

- DA (24) - podstawa czasu z DTE dla elementów nadawanych,
- DB (25) - podstawa czasu z DCE dla elementów nadawanych,
- DD (17) - elementowa podstawa czasu wytwarzana w DCE.

Pierwsza z nich sterowana jest przez DTE, pozostałe - przez DCE. Linie te umożliwiają:

- nadawanie przez DTE danych linią TxD w rytmie własnego zegara (z wykorzystaniem linii DA),
- nadawanie przez DTE danych linią TxD w rytmie zegara pochodzącego z DCE (z wykorzystaniem linii DB),
- odbieranie przez DTE danych z linii RxD w rytmie zegara DCE (z wykorzystaniem linii DD)

Przy nadawaniu danych przez DTE nie jest oczywiście możliwe wykorzystanie jednoczesne linii DA i DB.

Na liniach DA i DD przejście ze stanu "1", do stanu "0" powinno wskazywać środek bitu nadawanego odpowiednio przez DTE i DCE, i w ten sposób określać optymalny moment próbkowania bitu przez urządzenie odbierające. Na linii DB przejście ze stanu "0" do "1" powinno wskazywać chwilę, w której urządzenie nadające informację ma zacząć przekazywać nowy bit, czyli określać dla urządzenia nadającego rytm wysyłania bitów.

Wykorzystując linię DA do synchronizacji transmisji należy generować na niej sygnały zegarowe (na przemian "0" i "1" przez równe odcinki czasu) co najmniej wtedy, gdy aktywna jest linia DSR (6) informująca o gotowości DCE do współpracy z DTE. Korzystanie z linii DB i DD wymaga zapewnienia istnienia na nich sygnałów zegarowych w przypadku aktywnego stanu linii DCD, sygnalizującego poprawność sygnału odbieranego.

Linie synchronizacji zwykle nie są konieczne przy transmisji szeregowej między komputerem a przyrządem pomiarowym.

Linie masy

W interfejsie RS-232C rozróżnia się dwie masy:

- PG (1) - masa ochronna,
- SG (7) - masa sygnałowa.

Pierwsza z nich jest masą zabezpieczającą (ang. Protective Ground), łączoną z obudową urządzenia; druga - masą sygnałową (ang. Signal Ground), stanowiącą odniesienie dla wszystkich pozostałych sygnałów interfejsu.

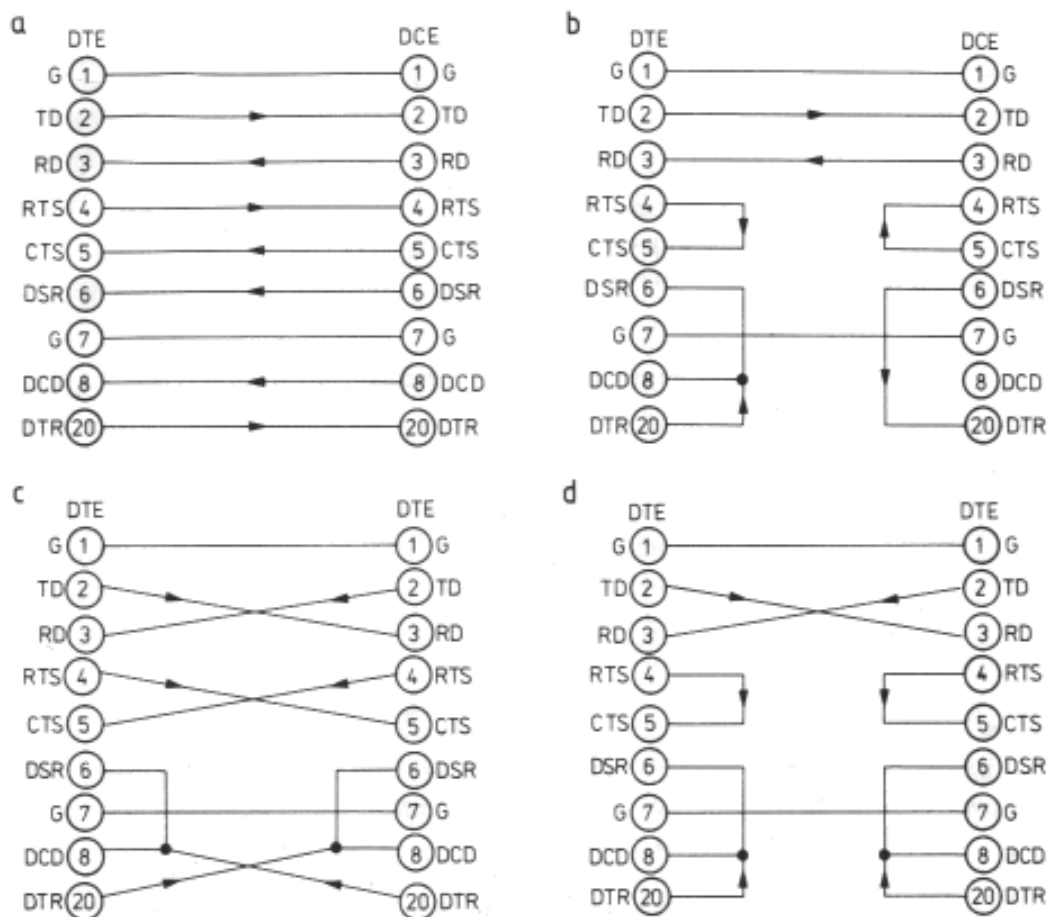
Łączenie urządzeń

Ujawniły się dwa podstawowe problemy związane z transmisją RS232C.

- a) Z punktu widzenia złącza zdefiniowano dwa rodzaje urządzeń w ten sposób, że wejścia urządzenia jednego rodzaju odpowiadają wyjściom urządzenia drugiego rodzaju i na odwrót.
- b) Zdefiniowano pięć sygnałów potwierdzenia (ang. handshaking signals). Niektóre urządzenia wysyłają je i oczekują odpowiedzi na te sygnały, natomiast inne ignorują sygnały handshakingu prowadzone do ich wejść i same ich nie wysyłają. Występuje tu jeszcze jeden istotny problem. RS232C wprowadzono w celu znormalizowania połączeń między urządzeniami typu DTE (data terminal equipment) i urządzeniami typu DCE (data communication equipment) Terminal jest zawsze urządzeniem DTE, natomiast modem jest zawsze urządzeniem DCE. Niestety, inne urządzenia, między innymi mikrokomputery, mogą być jednym i drugim. IBM PC jest urządzeniem typu DTE wyposażonym w męską część złącza, natomiast większość dużych komputerów zachowuje się jak urządzenie typu DCE. Połączenie DTE z DCE jest rozumiane jako złożenie ze sobą obu połówek złącza DB-25 (które mogą być zarówno męskie, jak i żeńskie, i to po obu stronach!). Przy odrobinie szczęścia może się zdarzyć, że łącze będzie działać. Ten łut szczęścia jest potrzebny ze względu na to, że nie wiemy, które linie handshakingu i w jaki sposób wykorzystuje jedno i drugie urządzenie. Oprócz problemów sprzętowych mogą tu jeszcze odegrać swoją rolę problemy z oprogramowaniem, tzn. nawet po pomyślnym rozwiązaniu problemu z kablem trzeba uzgodnić szybkość transmisji, sposób reakcji na bit parzystości itp. Przy łączeniu ze

sobą urządzeń tego samego typu nie można zwyczajnie złożyć obu połówek złącza razem, gdyż wtedy zostaną zwarte ze sobą wyjścia obu urządzeń: urządzenie typu DTE ma wyjście na wyprowadzeniu 2, a wejście na wyprowadzeniu 3 złącza, natomiast urządzenie typu DCE - odwrotnie. Dlatego trzeba zastosować specjalny kabel (nazywany "modemem zerowym"), w którym skrzyżowano przewody prowadzące do wyprowadzeń 2 i 3. Niestety, nie wyczerpuje to koniecznych w tym przypadku działań.

Gdyby wszystkie urządzenia pracujące według standardu RS232C wysyłały i odbierały cały komplet sygnałów, do poprawnej pracy łącza wystarczyłoby proste połączenie urządzenia DTE z urządzeniem DCE lub skrzyżowane połączenie urządzeń DTE z DTE lub DCE z DCE. Jeśli jednak połączy się urządzenie całkowicie ignorujące handshaking z urządzeniem oczekującym na sygnały potwierdzeń, transmisja danych nie jest możliwa. Tak więc trzeba przyjąć strategię postępowania odpowiadającą rzeczywistości, co czasami wymaga użycia podstępu. Na rysunku pokazano, w jaki sposób należy wykonać kable, które umożliwią transmisję w każdej sytuacji (ściślej mówiąc, prawie w każdej sytuacji). Połączenia przedstawione na rys. a dotyczą urządzeń DTE i DCE, wymagających pełnego handshakingu. RTS i CTS stanowią jedną parę linii potwierdzeń, a DTR i DSR drugą. Taki sam przypadek, dotyczący dwóch urządzeń z pełnym handshakingiem, lecz obu typu DTE, przedstawiono na rys. c. Tym razem trzeba użyć "modemu zerowego" w celu skrzyżowania odpowiednich linii sygnałowych dwu urządzeń tego samego typu. Kabel wykonany według tego rysunku będzie również odpowiedni do łączenia dwóch urządzeń typu DCE, przy czym dla zachowania porządku należałoby na rysunku odwrócić kierunki strzałek i usunąć połączenia z wyprowadzeniami 8. Kable te nie będą pracować w przypadku, gdy jedno urządzenie wymaga doprowadzenia sygnałów potwierdzających, a drugie ich nie wysyła. Najprostszym sposobem rozwiązania problemu sygnałów potwierdzających jest takie wykonanie kabla, aby każde urządzenie wysyłało i odbierało własne sygnały handshakingu, czyli aby samo sobie zezwalało na transmisję. Jak to należy zrobić, pokazano na rys. b dla pary urządzeń DTE i DCE oraz na rys. d dla pary urządzeń DTE, DTE (taki kabel jest również dobry dla pary DCE, DCE, lecz należy usunąć połączenia z wyprowadzeniami 8).



Tryby pracy

Wyróżnić możemy 3 tryby pracy:

- Simpleks - transmisja odbywa się tylko w jednym kierunku,
- Półdupleks - transmisja odbywa się w obu kierunkach na przemian,
- Dupleks (Pełny dupleks) - transmisja odbywa się w obu kierunkach jednocześnie.

Kontrola przepływu danych

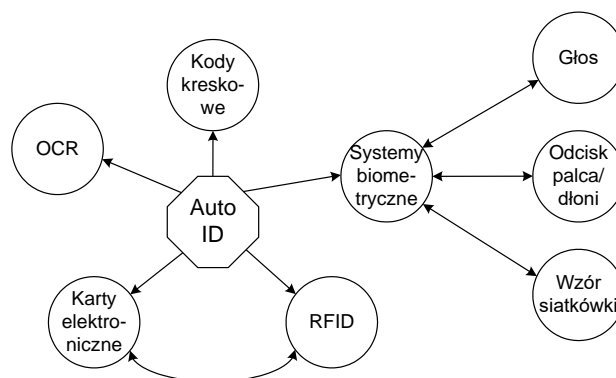
Kontrolę przepływu możemy podzielić na :

- Sprzętową - za pomocą sygnałów RTS/CTS - urządzenie, które nie może chwilowo odebrać danych (np. ma pełny bufor odbiorczy) dezaktywuje sygnał CTS, co powoduje zatrzymanie transmisji kolejnych znaków ,do czasu ponownego przejścia sygnału CTS w stan aktywny. Wykorzystywane są wszystkie opisane sygnały.

- Programową - protokół XON/XOFF - urządzenie aby wstrzymać lub zatrzymać transmisję wysyła do urządzenia transmitującego specjalny znak sterujący. Przy takim rozwiązaniu do transmisji wystarczą tylko linie TxD i RxD, lecz transmisja musi się odbywać w trybie pełnego duplexu.

3.2 Układy identyfikacji bezstykowej RFID

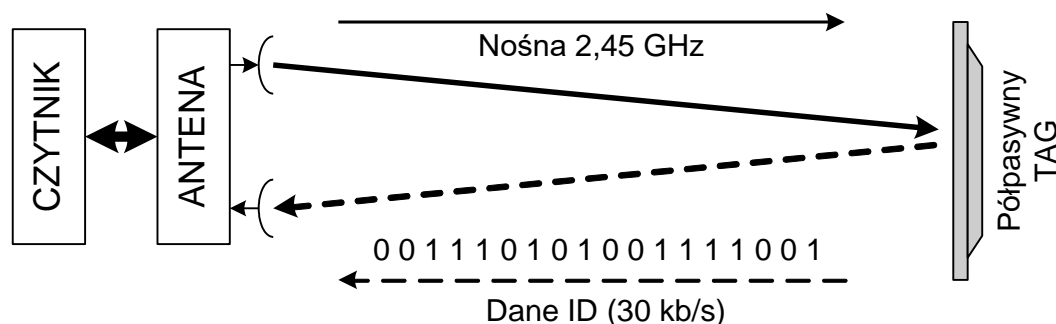
Spośród wielu nowych technik i technologii, które w ostatnich latach wpłynęły na rozwój systemów telematiki transportu, znaczące miejsce zajmują procedury automatycznej identyfikacji (ang. *Auto-ID*). Ich olbrzymia różnorodność aplikacyjna, powiązana z bogatą ofertą sprzętową (rys. 3.1), pozwala na znajdowanie wciąż nowych obszarów zastosowań. Dla transportu głównymi obszarami aplikacyjnymi są tu przede wszystkim systemy identyfikacji pojazdów oraz elektronicznej płatności. Wśród nich na szczególną uwagę zasługują układy identyfikacji bezprzewodowej RFID (ang. *Radio Frequency Identification*). Na korzyść układów RFID przemawia przede wszystkim ich główna właściwość – zdolność do zdalnej (bezprzewodowej i bezstykowej) identyfikacji, która w praktyce bezpośrednio uwzględnia podstawową cechę transportu - mobilność pojazdów. Dzięki temu realizacja wielu funkcji systemowych (np. wjazd na strzeżony parking) stała się możliwa bez absorbowania uwagi kierowcy.



Rys. 3.1 Główne obszary zastosowań automatycznej identyfikacji

W wielkim uproszczeniu zasada działania układów RFID opiera się na modulacji sygnału częstotliwości nośnej (wysyłanego z anteny czytnika) w sposób umożliwiający zapisanie w tym sygnale danych cyfrowych, zawartych w samym układzie (rys. 3.2). Tak zmodulowany sygnał wraca następnie do anteny, gdzie czytnik dekoduje zapisane dane. W

początkowym okresie transmitowanymi danymi był jedynie unikalny numer układu RFID. Pozwalał on na jego jednoznaczną identyfikację w sposób ideowo identyczny, jak w przypadku kart elektronicznych (inteligentnych, chipowych), tyle tylko, że na drodze bezprzewodowej. Obecnie, dzięki postępowi technologicznemu, istnieje dość duża swoboda wprowadzania (programowania) do układu RFID obok unikalnego numeru identyfikacyjnego, własnych danych, np. kodu producenta lub systemu, numeru seryjnego, itp.



Rys. 3.2 Zasada działania układów RFID

Dalszy rozwój układów RFID zaowocował w ostatnim czasie wprowadzeniem układów dalekiego zasięgu (od 8 do 10 metrów), tzw. pół-pasywnych tagów, wyposażonych we własny układ zasilania i aktywowany polem/sygnałem w.cz. W rozwiązaniu tym ważny jest nie tylko znaczny zasięg, w dodatku przy szybkościach do 100 km/h, ale także fakt, że zasilanie taga nie wymaga dodatkowej instalacji (funkcja baterii) i wystarcza do ośmiu lat użytkowania. Dzięki temu układy RFID wyszły z obszarów prostej, bezpośredniej identyfikacji pojazdów i wkroczyły do systemów zarządzania ruchem, w tym także w odniesieniu do pojazdów komunikacji miejskiej. Przykładem jest tutaj system zarządzania w obrębie dworca autobusowego w Tuluzie, miasta będącego największym węzłem komunikacyjnym południowo-zachodniej Francji. Nie brakuje też zastosowań w nadzorze ładunków transportu kolejowego.

Standardy RFID

O ile zasada działania kart i sposób transmisji informacji jest w przypadku wszystkich rozwiązań taki sam, o tyle metoda kodowania, a także same przesyłane dane i możliwości komunikacyjne poszczególnych standardów kart są bardzo różnorodne. Nie sposób wymienić tu wszystkie z nich, dlatego to kilka najpopularniejszych:

UNIQUE 125kHz

To najprostszy ze stosowanych standardów kart. Częstotliwością nośną jest częstotliwość 125kHz, a karta przechowuje jedynie swój 40-sto bitowy numer seryjny programowany podczas produkcji w fabryce. Teoretycznie numery kart nie powinny się powtarzać, ale na dzień dzisiejszy istnieją na świecie karty o dublujących się numerach.

Q5

Karty podobne do UNIQUE 125 kHz posiadające dodatkową pamięć EEPROM możliwą do wielokrotnego programowania przez użytkownika (także dzięki modulacji fal radiowych, tym razem przez antenę nadawczo-odbiorczą urządzenia), jednak rozwiązanie to jest stosunkowo rzadko spotykane. Pamięć w takich kartach jest chroniona hasłem – bez jego podania jej odczyt lub modyfikacja są niemożliwe.

Hitag

Standard także wykorzystujący częstotliwość nośną 125kHz i udostępniający pamięć EEPROM, jednak posiadający bardziej rozbudowane możliwości w stosunku do wcześniej wymienionych kart – np. możliwość szyfrowania danych i system antykolizyjny (w przypadku transponderów UNIQUE i Q5 jeśli w polu czytnika umieścimy 2 karty, to żadna informacja nie zostanie odczytana – dane z obu kart będą się zakłócać, tymczasem standard HITAG, dzięki systemowi antykolizyjnemu umożliwia odczyt wielu znaczników umieszczonych w polu czytnika i indywidualną komunikację z każdym z nich)

Mifare

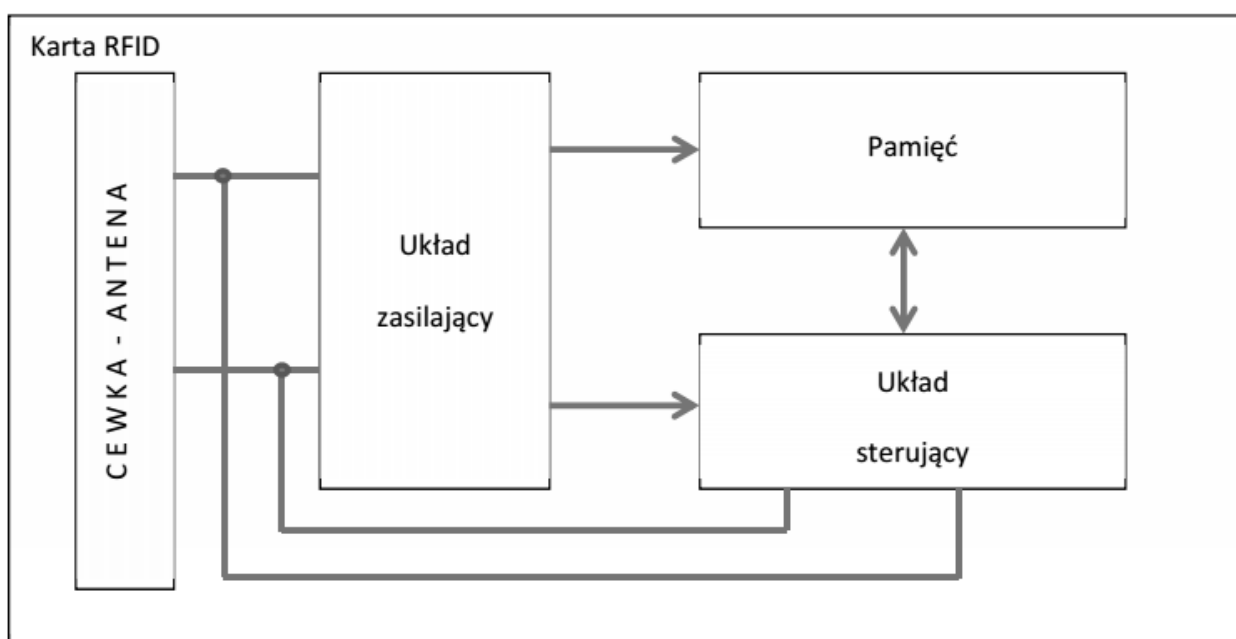
Częstotliwość nośna to 13,56MHz, jednak w tym standardzie mamy już dostępny szeroki wachlarz możliwości „wewnętrznych” karty, który oferuje między innymi system uwierzytelniania połączenia, szyfrowanie przesyłanych danych i wykonywanie bardziej skomplikowanych operacji. Standard ten stosowany jest między innymi w zbliżeniowych kartach płatniczych, kartach (biletach okresowych) komunikacji miejskiej niektórych miast a także droższych systemach kontroli czasu pracy czy dostępu.

Pomimo rozwoju technologicznego podstawowym elementem systemów zbliżeniowych pozostaje jeszcze klasyczny (już) transponder, występujący najczęściej w postaci karty zbliżeniowej standardu Unique. Są to cienkie kawałki białego, elastycznego plastiku, rozmiarami zbliżone z typową kartą płatniczą, z zatopioną wewnątrz anteną i układem scalonym (chip'em). Każdy transponder ma unikalny kod, który jest wysyłany do czytnika, gdy znajdzie się w jego polu. Karty Unique są kartami pasywnymi (nie posiadają

własnego zasilania - baterii). Na samych kartach można wykonywać nadruki przy pomocy specjalnych drukarek termotransferowych.

Budowa karty RFID

Karta, czy dowolny inny tag RFID składa się z 2 ważnych elementów: cewki (pełniącej jednocześnie rolę elementu zasilającego układ oraz nadawczo-odbiorczego) oraz układu scalonego odpowiedzialnego za funkcjonowanie karty (rys. 3.3). Przyglądając się całemu systemowi można by narysować następujący uproszczony schemat blokowy naszej karty (odtąd dla jasności będę już używał tego określenia dla wszystkich tagów RFID):



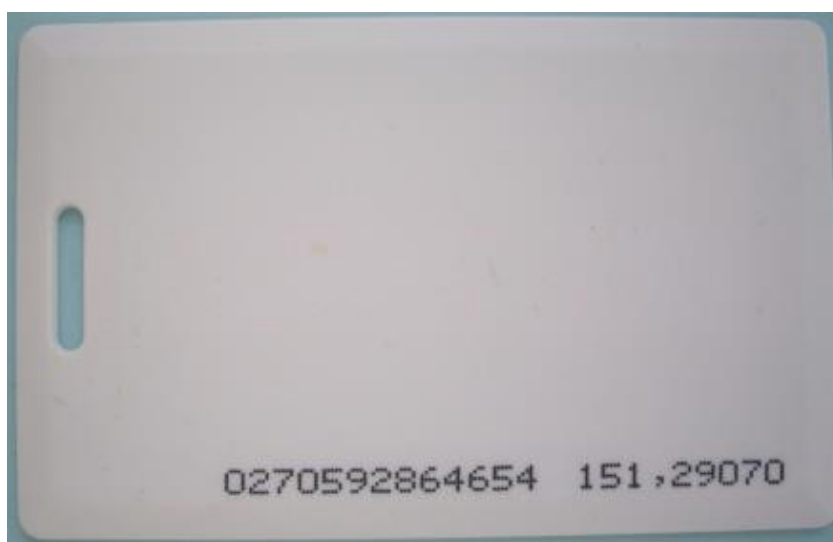
Rys. 3.3 Budowa karty RFID

Zasada działania układu jest prosta i zgodna z tym co przedstawiono wcześniej:

1. Zmienne pole magnetyczne indukuje w cewce prąd, który jest wykorzystywany do naładowania małych kondensatorów na tyle, by energii starczyło do wykonania niezbędnych operacji.
2. Poprawnie zasilony układ sterujący może komunikować się z pamięcią, i jednocześnie badać sygnał pochodzący z cewki (np. na jego podstawie generować sygnał zegarowy taktujący wykonywanie kolejnych operacji i nadawanie kolejnych bitów, czy odbierać dane z czytnika/programatora) oraz wpływać na parametry całego obwodu w celu odpowiedniego zaburzenia pola magnetycznego i nadania danych.

W omawianych przez nas szczegółowo kartach UNIQUE układ sterujący na podstawie sygnału częstotliwości nośnej generuje sygnał taktujący nadawanie, a poprzez załączanie lub odłączanie od obwodu cewki dodatkowej rezystancji zmienia natężenie prądu przez nią płynącego, a co za tym idzie zmienia natężenie pola magnetycznego przez nią generowanego (im większy płynie prąd tym pole większe). Z kolei silniejsze pole generowane przez cewkę karty powoduje osłabienie pola czytnika i w efekcie spadek amplitudy sygnału elektrycznego w jego cewce – taki stan jest odczytywany jako logiczna 1. Z kolei jeśli pole generowane przez kartę jest słabsze, nie następuje wyraźny spadek amplitudy sygnału nadajnika i odbieramy logiczne 0.

W rzeczywistości karta RFID wygląda tak (rys. 3.4 i 3.5 na przykładzie karty RFID UNIQUE 125kHz w obudowie ClamShell):



Rys. 3.4 Przednia strona obudowy karty RFID



Rys. 3.5 Karta ze zdemontowaną tylną częścią obudowy – widoczna cewka i układ scalony

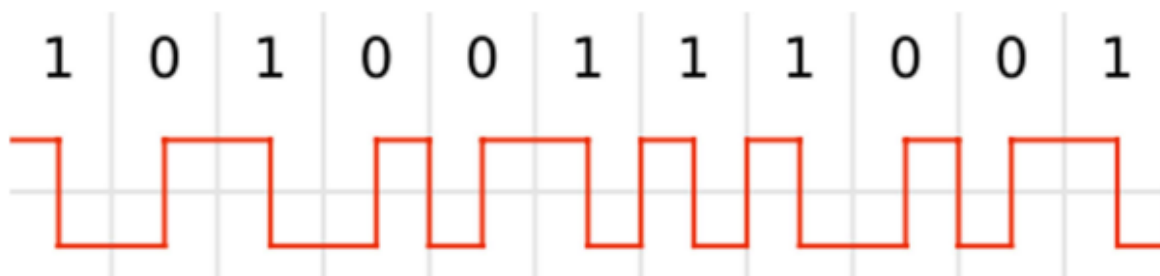
Kodowanie danych na karcie RFID

W celu zakodowania (przetworzenia do postaci dogodnej dla odczytu, nie mulić z szyfrowaniem) danych na karcie RFID wykorzystywany jest Kod Manchester. Jest kod liniowy sygnału cyfrowego (modulacja bifazowa). Na początku sygnał przyjmuje stan odpowiadający jego wartości binarnej, w środku czasu transmisji bitu następuje zmiana sygnału na przeciwny; dla zera z niskiego na wysokie, dla jedynki – z wysokiego na niski. Modulacja po raz pierwszy została wprowadzona przez G. E. Thomasa w 1949 r.

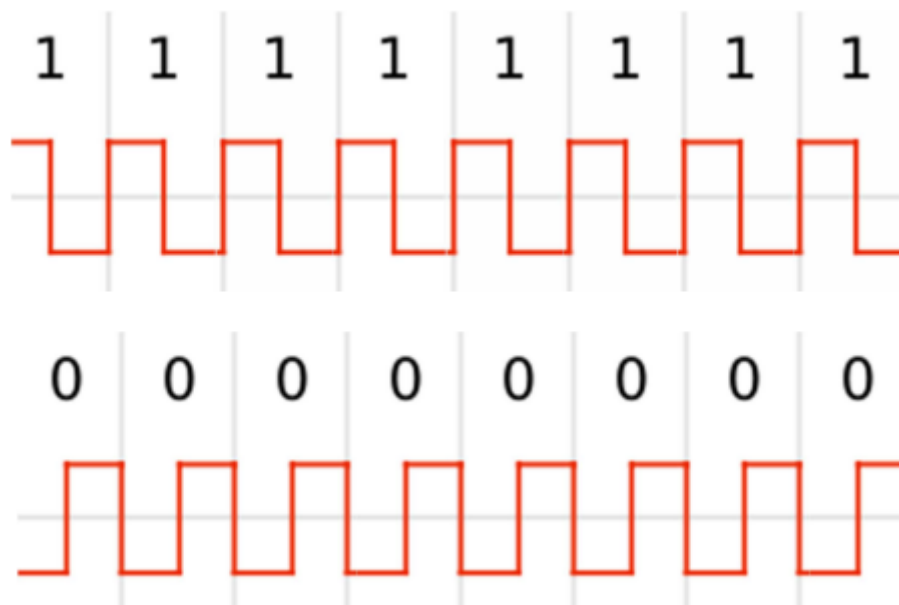
Czyli jeśli rozważymy nadawanie jednego bitu to dla nadawania 1 w połowie cyklu występuje zbocze opadające, a dla nadawania 0 – zbocze narastające. Ilustrują to poniższe rysunki:



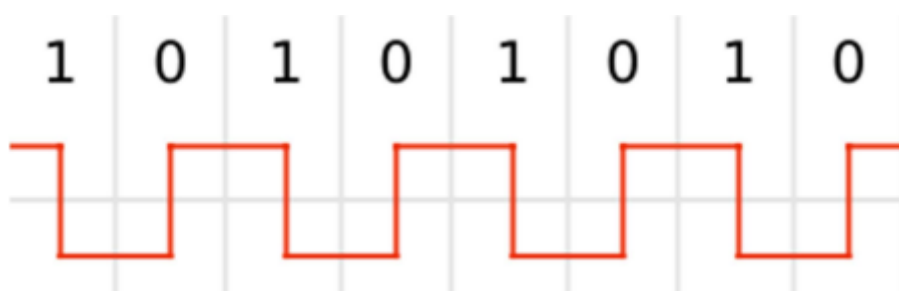
Jeżeli teraz spojrzymy na większy fragment danych nadawanych z wykorzystaniem tego kodu to zauważymy jego bardzo ciekawą właściwość, pomagającą w jego zdekodowaniu:



Otóż jeśli nadawane są ciągi takich samych bitów (jedynki lub zera) to czas między kolejnymi zboczami zawsze jest równy połowie okresu (w naszym przypadku będą to 32 cykle zegarowe).



Z kolei tylko i wyłącznie w momencie zmiany nadawanego bitu (z 0 na 1 lub z 1 na 0) czas między kolejnymi zboczami jest równy pełnemu okresowi (czyli 64 cykom zegara):



Dekodowanie takiego sygnału będzie, zatem dość proste. Na początku przyjmiemy jakąś wartość nadawanego bitu, np. $B=1$ (niestety, karty RFID nadają dane ciągle, „w kółko”, nie stosując żadnej zależności czasowej dla wskazania początku nadawania pakietu danych) i postępujemy według algorytmu:

1. Jeśli czas między zboczami wynosi 32 cykle to:
 - a. zwiększ licznik zboczy o 1
 - b. jeśli to parzyste zbocze to zapisz odebrany bit o wartości B
2. Jeśli czas między zboczami wynosi 64 cykle to:
 - a. zwiększ licznik zboczy o 2
 - b. zmień wartość B na przeciwną
 - c. zapisz odebrany bit o wartości B

W taki oto sposób bardzo szybko możemy zdekodować całą ramkę.

Ramka danych RFID

Niestety, jak już wcześniej wspomniano, karty nadają swoje dane w pętli nie stosując żadnego czasowego wyróżnika dla sygnalizowania rozpoczęcia transmisji danych, co znacznie utrudnia odbiór danych. Z tego powodu jedyną możliwością odnalezienia początku naszych danych jest poszukiwanie wśród odebranych dowolnych 64 bitów pierwszego wystąpienia 9-ciu kolejnych bitów o wartości 1 (rys. 3.6). Dlaczego jednak taka kombinacja w 100% informuje o początku ramki? To bardzo proste. Dzięki bitom parzystości w żadnym innym miejscu nie może wystąpić pod rząd więcej niż 8 bitów o wartości 1. Dalej przed bitami startu zawsze nadawany jest bit stopu poprzedniej ramki o wartości 0, a jeśli dane będziemy zawsze przeglądać podążając w prawo to pierwsze wystąpienie 9-ciu jedynek pod rząd daje nam 100% pewność, że natrafiliśmy na nagłówek ramki.

S	S	S	S	S	S	S	S	S	S – 9 bitów startu – zawsze mają wartość 1				
8 bitów – kod producenta karty				D00	D01	D02	D03	P0	P0...P9 – bity parzystości (<i>even parity</i>) dla każdego z wierszy tabelki.				
				D04	D05	D06	D07	P1					
32 bity – unikalny kod karty				D08	D09	D10	D11	P2	Jeśli w danym wierszu znajduje się nieparzysta ilość jedynek to bit przyjmuje wartość 1, w przeciwnym wypadku przyjmuje on wartość 0.				
				D12	D13	D14	D15	P3					
				D16	D17	D18	D19	P4					
				D20	D21	D22	D23	P5					
								D24	D25	D26	D27	P6	CP0...CP3 – bity parzystości (<i>even parity</i>) dla każdej z kolumn tabelki. Działają na takiej samej zasadzie jak parzystość wierszy.
								D28	D29	D30	D31	P7	
								D32	D33	D34	D35	P8	
								D36	D37	D38	D39	P9	
				CP0	CP1	CP2	CP3	S0	S0 – bit stopu – zawsze ma wartość 0				

Rys. 3.6 Ramka danych karty RFID

To jednak nie wszystko. Z powodu braku czasowego oznaczenia początku ramki oprogramowanie może nie tylko rozpocząć odbiór od dowolnego miejsca pakietu danych (co będzie się zdarzać w 99% przypadków), ale także może rozpocząć dekodowanie kodu Manchester od złego zbocza. W wyniku tego (co też możecie sprawdzić samodzielnie rysując przebieg powstały w wyniku zakodowania jakichś danych i potem rozpoczynając dekodowanie od 2-giego zbocza) otrzymany ciąg bitów będzie zanegowany. Ponadto, żeby nie było nam zbyt łatwo, transmisja może przebiegać z błędami, o czym będzie nas informować wartość bitu parzystości (mamy błąd w transmisji, jeśli odczytana wartość bit parzystości jest różna od wyliczonej na podstawie danych).

Prosty algorytm dekodowania danych RFID

1. Przesuwaj dane dopóki nie natrafisz na bity startu. Jeśli nie znajdziesz bitów startu przejdź do kroku 4
2. Sprawdzamy parzystość pionową i poziomą. Jeśli coś się nie zgadza to przejdź do kroku 4
3. Odczytaj dane z ramki i złóż je w pełne bity. Prześlij dane gdzie trzeba i zakończ działanie programu.
4. Zaneguj całe dane.
5. Przesuwaj dane dopóki nie natrafisz na bity startu. Jeśli nie znajdziesz bitów startu przejdź do kroku 8
6. Sprawdzamy parzystość pionową i poziomą. Jeśli coś się nie zgadza to przejdź do kroku 8
7. Odczytaj dane z ramki i złóż je w pełne bity. Prześlij dane gdzie trzeba i zakończ działanie programu.
8. Nie odebrano prawidłowych danych – spróbuj dokonać ponownie odbioru danych.

Bezpieczeństwo kart RFID

W teorii karty standardu UNIQUE są dosyć bezpieczne – numery powinny być niepowtarzalne, a karty równie bezpieczne jak klucze. W praktyce niestety tak nie jest. Po pierwsze unikalne numery kart skończyły się już jakiś czas temu (powstało więcej niż 2^{32} kart) i powstają karty o dublujących się numerach. Jednak, jeśli zastanowimy się dobrze ta wada tego systemu nie jest bardzo groźna – prawdopodobieństwo, że w danym rejonie świata znajdą się 2 karty o takich samych numerach jest znikomo małe a ich celowe odnalezienie jest w zasadzie niemożliwe.

Poważny problem stwarza jednak możliwość stosunkowo łatwego kopiowania takich kart. Jak się łatwo przekonamy szukając dłużej w sieci znajdziemy parę bardziej lub mniej skomplikowanych projektów budowy różnego rodzaju klonerów i emulatorów kart. Z powodu braku jakichkolwiek bardziej zaawansowanych metod zabezpieczenia w celu skopiowania karty wystarczy odczytać jej numer seryjny (co czasem można dokonać dysponując odpowiednim sprzętem z odległości paru metrów), a następnie zbudować układ wysyłający te dane do czytnika. Ciekawszy z projektów <http://scanlime.org/2008/09/using-an-avr-as-an-rfid-tag/> wykorzystuje jedynie odpowiednią cewkę i malutki mikrokontroler ATTiny85 – w wersji SMD takowym procesorem można nawet zastąpić oryginalny układ

scalony w karcie i posiadać dzięki temu sklonowaną kartę wyglądającą zupełnie jak oryginalna.

Czytnik kart

Obecnie większość czytników kart RFID udostępnia nam dane odczytane ze zbliżonego tagu za pomocą interfejsu RS232. Rozwiązanie takie jest bardzo proste w obsłudze i nie sprawia prawie żadnych trudności programistycznych – wystarczy tylko podpiąć czytnik do komputera lub procesora, ustawić odpowiednio parametry komunikacji szeregowej i odbierać zdekodowane dane.

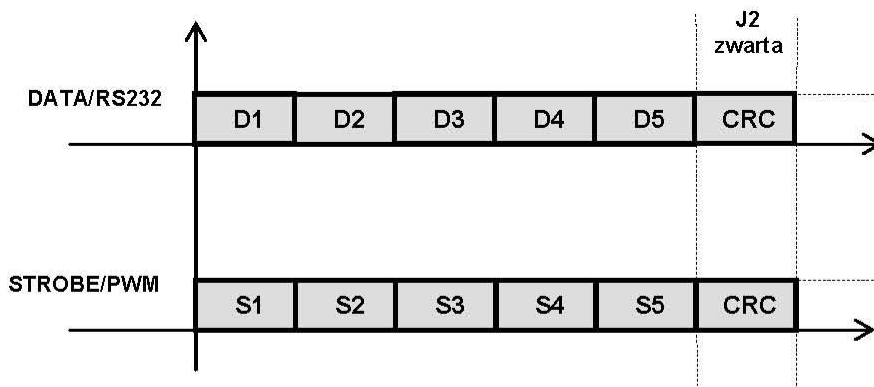
Przykładem prostego czytnika kodów transponderów UNIQUE jest prezentowany na ćwiczeniu czytnik HB2300. Wykorzystuje on niezależny moduł odbiorczy TRD-80 CLASSIC, który jest faktycznym czytnikiem (w sensie odbioru) transponderów. Pozwala on na szybkie i łatwe konstruowanie urządzeń wymagających bezstykowej identyfikacji. Różnorodność jego formatów danych wyjściowych umożliwia łatwe sprzężenie czytnika z różnymi systemami mikroprocesorowymi. Planarna cewka/antena wytrawiona na tej samej płytce, co mikrokontroler, zapewnia wysoką stabilność i powtarzalność parametrów. Szeroki zakres napięć zasilających pozwala na bezpośrednią integrację modułu z układami zasilanymi napięciem od 3V do 6V.

Parametry techniczne modułu odbiorczego TRD-80 CLASSIC:

- zasilanie od 3V do 6V,
- zintegrowana antena,
- 4 formaty danych wyjściowych,
- wyjście BEEP,
- wyjście PRESENT,
- zasięg odczytu do kilkunastu cm.

Opis wyprowadzeń modułu odbiorczego TRD-80 CLASSIC:

Pin	Nazwa	Funkcja
1	GND [CZARNY]	Masa.
2	DALLAS / INP0 (opcja!)	WYJŚCIE – emulacja pastylki DALLAS DS1990A. Na tym pinie można odczytać kod transpondera w sposób identyczny z odczytem kodów z pastylki DALLAS DS1990A. Pozwala to w prosty sposób podłączyć czytnik do urządzeń przystosowanych do współpracy z pastylkami DALLAS. Wprowadzenie transpondera w pole czytnika jest równoważne z przytknięciem pastylki do zestyków.
3	DATA / RS232 [ŻÓŁTY]	WYJŚCIE – 40 bitów danych (5 bajtów) wysyłanych w formacie RS232-TTL. Kod transpondera generowany na tym pinie można odczytywać wykorzystując standardowy asynchroniczny odbiornik kanału szeregowego (UART). W układach, w których użycie odbiornika UART jest niemożliwe można odczytywać poziomy poszczególne bity generowanych na tym pinie wykorzystując do strobowania impulsy pojawiające się na pinie 4.
4	STROBE / PWM [NIEBIESKI]	WYJŚCIE – ciąg czterdziestu impulsów ‘0’ odpowiadających czterdziestu bitom kodu transpondera. Długość poszczególnych impulsów zależy od wartości kodowanego bitu i jest równa 30µs dla „1” i 120µs dla „0”. Początek każdego impulsu jest opóźniony o 120µs w stosunku bitów wysyłanych na pinie 3 (RS232-TTL). Impulsy generowane na tym pinie mogą być użyte do odczytu kodu transpondera poprzez pomiar ich długości lub jako impulsy strobowujące wartości bitów odczytywane z pinu 3.
5	BEGIN / BEEP [CZERWONY]	WYJŚCIE – pojedynczy impuls „0” o długości 120µs i wyprzedzający o 120µs każdą paczkę danych.
6	Vcc	Zasilanie +5V.

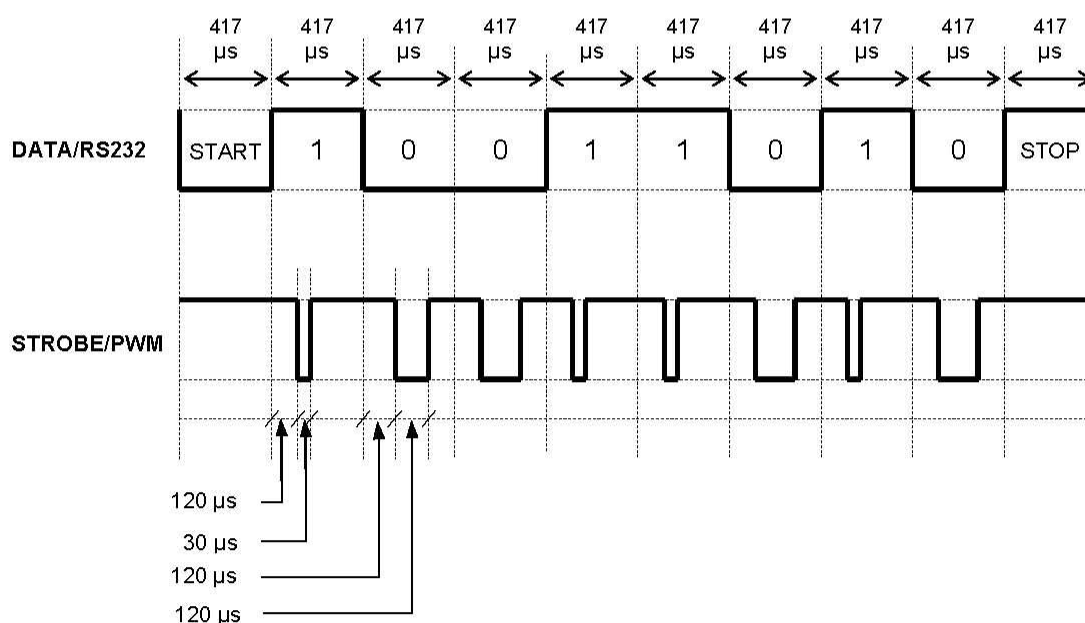


Na pinie DATA/RS232 pojawia się pięć bajtów kodu transpondera w formacie RS232-TTL plus opcjonalnie suma kontrolna. Na pinie STROBE/PWM pojawiają się impulsy strobujące. Pojawiają się one tylko wtedy, kiedy na pinie DATA/RS232 przesyłane są bity danych (bity START i STOP przepuszczane są bez strobów). Szerokość impulsów strobujących zmienia się w zależności od wartości przesyłanego bitu.

$CRC = D1 \wedge D2 \wedge D3 \wedge D4 \wedge D5$ (suma XOR), wysyłana tylko jeśli J2 jest zwarta.

Prędkość 2400 bit/s.

Format jednego bajtu wysyланego na pinach DATA/RS232 i STROBE/PWM (np. bajt 0x59):



Transmisja każdego bajtu zaczyna się od najmłodszego bitu.

Kod transpondera możemy odczytywać na 3 różne sposoby:

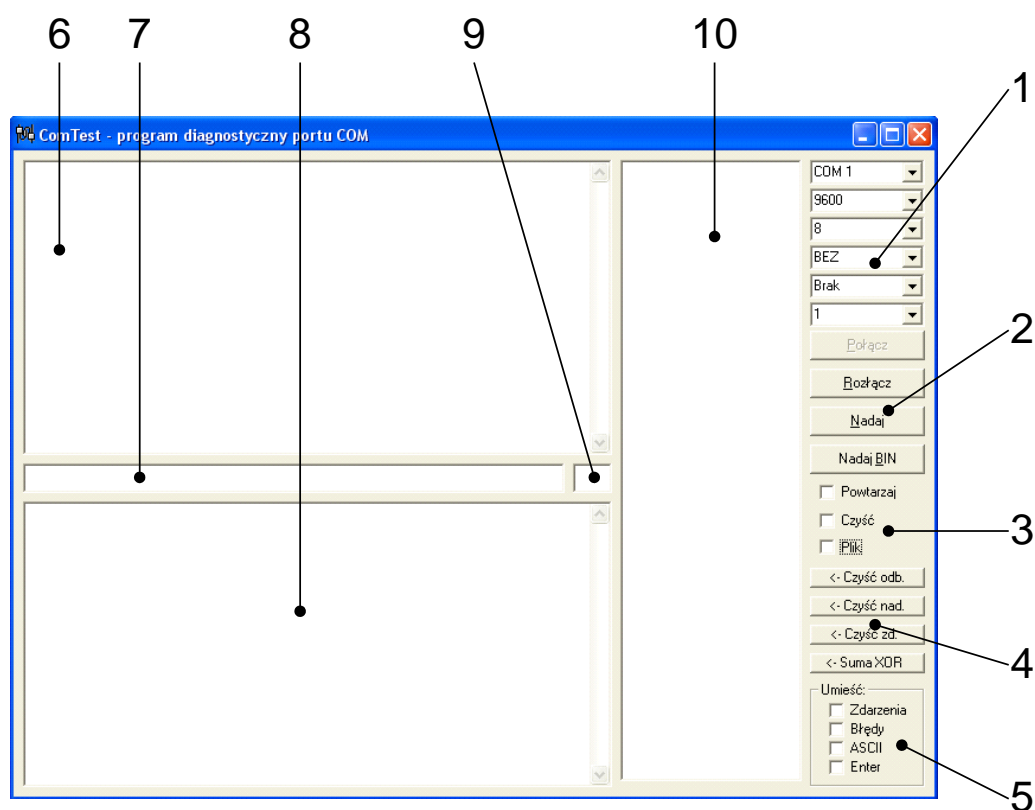
- a) korzystając tylko z pinu DATA/RS232 - format RS232-TTL
- b) wykorzystując DATA/RS232 jako bity danych, a STROBE/PWM jako impulsy strobujące
- c) wykorzystując STROBE/PWM jako źródło impulsów z modulowaną szerokością (PWM)

Rys. 3.3 Przykładowe przebiegi czasowe na pinach DATA/RS232 i STROBE/PWM

4. Program narzędziowy ComTest

Jego zadaniem jest obsługa dostępnych w Windows portów COM (RS232C) komputera laboratoryjnego. Po uruchomieniu programu widoczne jest jego standardowe okno (rys. 4.1), którego organizacja nie ulega zmianie podczas pracy.

- 1 – pole wyboru portu i parametrów transmisji,
- 2 – klawisze poleceń operacji na porcie,
- 3 – pola wyboru opcji operacji na porcie,
- 4 – klawisze operacji na polach tekstowych programu,
- 5 – pola wyboru opcji dodatkowych informacji,
- 6 – pole danych odbieranych,
- 7 – pole danych do wysłania lub nazwa pliku,
- 8 – pole danych nadanych,
- 9 – pole współczynnika skali czasu,
- 10 – pole zdarzeń i komunikatów



Rys. 4.1 Program diagnostyczny i obsługi portów RS232C ComTest

Dostęp do danego portu uzyskuje się po wybraniu jego numeru z pierwszej listy (COM 1, COM 2) oraz jego konfiguracji do której należą kolejno: wybór szybkości pracy (300, ... 9600, ...), ilości bitów danych (5, 6, 7, 8), rodzaj kontroli parzystości (brak, parzysty, nieparzysty), sposobu sterowania przepływem danych (brak, xon/xoff, ...) oraz ilości bitów stopu (1 lub 2). Najczęstsze ustawienie wykorzystywane podczas ćwiczeń to: COM n (gdzie n zależy od zestawionego połączenia Bluetooth), 4800, 8, bez kontroli parzystości, brak sterowania przepływem, 1 bit stopu. Po dokonaniu konfiguracji należy nacisnąć klawisz *POŁĄCZ* i program otwiera dostęp do wybranego portu. W przypadku błędu, np. z powodu zajętości portu przez inną aplikację, program wyświetli odpowiedni komunikat. Zamknięcie portu następuje po naciśnięciu klawisza *ROZŁĄCZ*. W polu nr 6 pokazywane są wszystkie znaki odbierane z otwartego portu COM. Należy jednak pamiętać, że nie wszystkie znaki ASCII będą widziane w czytelnej postaci. Pole nr 7 służy do bezpośredniego wysyłania danych do portu COM lub określenia nazwy pliku do odczytu lub zapisu. W celu wysłania danych należy wpisać w polu nr 7 odpowiednią sekwencję znaków i nacisnąć Enter lub klawisz *NADAJ*. W przypadku zaznaczenia opcji *POWTÓRZ* program będzie „w kółko” nadawał zapisaną sekwencję. Jeżeli umieścimy tam nazwę pliku i zaznaczymy opcję *PLIK* (domyślnym katalogiem jest c:\gps\), to program wyśle do portu zawartość tego pliku. Przy zaznaczonej opcji *POWTÓRZ* wysyłanie będzie realizowane „w kółko”. Jeżeli zaznaczono opcję *CZYŚĆ*, to po wysłaniu sekwencji nastąpi skasowanie zawartości pola nr 8, gdzie umieszczane są wszystkie znaki wysłane do portu COM. Pole nr 10 służy natomiast do wyświetlania zachodzących w porcie COM zdarzeń i informowania o wykrytych błędach transmisji. Możliwe jest to po zaznaczeniu odpowiedniej opcji w polu *UMIEŚĆ*. Zaznaczenie tam opcji *ASCII* spowoduje wyświetlenie w polu nr 6 nie znaków, lecz ich kodów ASCII, dla lepszej czytelności pomiędzy znakami „,◇”. Opcja *ENTER* powoduje dodanie do każdej wysyłanej do portu sekwencji znaków kodów nowej linii CR/LF, czyli 13 i 10. Kasowanie zawartości pól nr 6, 7 i 8 jest możliwe dzięki trzem klawiszom *CZYŚĆ*, odpowiednio *ODB.*, *NAD.*, *ZD*. Klawisz *SUMA XOR* służy do obliczenia sumy kontrolnej ramki danych protokołu NMEA, umieszczonej w polu nr 7.

Program może zapisywać odbierane dane bezpośrednio do pliku tekstowego. Podczas operacji na plikach ważna jest jednak kolejność wykonywanych działań:

- przy zapisie do pliku najpierw wpisujemy nazwę pliku w polu nr 7, zaznaczamy opcję *PLIK* i dopiero (po konfiguracji) otwieramy port poleceniem *POŁĄCZ*,
- przy odczycie danych z pliku najpierw odznaczamy opcję *PLIK* i dopiero (po

konfiguracji) otwieramy port poleceniem *POŁĄCZ*, następnie wpisujemy nazwę pliku w polu nr 7 i zaznaczamy opcję *PLIK*.

UWAGA!!! Program można uruchamiać wielokrotnie i obsługiwać kilka portów (urządzeń) jednocześnie.

5. Wykaz użytych skrótów i oznaczeń

Dla zwiększenia przejrzystości wprowadzono poniższe skróty oraz oznaczenia graficzne, które zostały wykorzystane w tekście instrukcji:



- zapisz przebieg na dysku,




- pytanie, na które odpowiedź musi znaleźć się w sprawozdaniu,

6. Uwagi praktyczne

Pod żadnym pozorem nie należy „upraszczać” ćwiczenia i próbować wykonywać jego części w sposób łączny lub w zmienionej kolejności. Jest to najszybsza droga do pomyłki w przeprowadzonych pomiarach i obserwacjach, co skutkuje odrzuceniem sprawozdania.

7. Przebieg ćwiczenia

Łącze RS232C

- 7.1 Zapoznaj się z budową stanowiska laboratoryjnego. Odszukaj i zidentyfikuj wszystkie elementy jego wyposażenia oraz przyrządy pomiarowe.
- 7.2 Odszukaj na stole laboratoryjnym *Kabel portu szeregowego*, który jest podłączony do portu **COM1**. Zapamiętaj tę wartość. 
- 7.3 Do *Kabla portu szeregowego* podłącz kanał pierwszy **CH1** przyst. oscyloskopowej **PO**, używając kabla koncentrycznego, czerwony. Tak podłączasz przystawkę oscyloskopową do masy sygnałowej **SG** i linii transmisji danych **TxD** portu RS232C.
- 7.4 Uruchom program ComTest. Wybierz kolejno z rozwijanych list: port **COM1**, szybkość transmisji (przepływność) 9600, 8 bitów danych, BEZ parzystości, brak sterowania przepływem, 1 bit stopu. Naciśnij klawisz *Połącz*. W pole nadawania (wąskie poziome) wpisz jeden znak – ostatnia cyfra z numeru indeksu pierwszego członka zespołu. Zaznacz opcję *Powtarzaj* i naciśnij *Nadaj*. Od tej chwili na linii **TxD** pojawia się ciągle nadawany znak.

- 7.5 Uruchom **RUN** przetwarzanie sygnału na przystawce oscyloskopowej **PO**. Wyłącz kanał drugi **CH2**. Ustaw {tryb **DSO**; **T/D** = 0,2ms; **V/D** = 10 V/dz; wyzwalenie **ON CH1**}. Ustaw poziom wyzwiania, by zapewnić odświeżanie przebiegu. Jeżeli przebieg nie mieści się na ekranie przystawki (jest „za wysoki”) to przesun go suwakiem w taki sposób, by była widoczna górna część, czyli zmiany poziomu impulsów. 🖱️
- 7.6 Zidentyfikuj bit startu. Pamiętaj, że obowiązuje tu logika ujemna („1” to poziom niski, a „0” to wysoki). 🖱️ Jaki jest czas trwania pojedynczego bitu? Jaka jest zależność między czasem trwania pojedynczego bitu a przepływnością? Jaka jest różnica między ustawioną przepływnością a wyliczoną z czasu trwania pojedynczego bitu? Skąd ta różnica? ❓
- 7.7 Pamiętaj, że 3-4 takie same bity łatwo pomylić z 2-3 – transmisja jest bez powrotu do zera. Wykorzystaj pionowe markery, którymi określałeś czas trwania pojedynczego bitu, do zliczania kolejnych takich samych bitów. 🖱️ Jaki jest binarny zapis wysyłanego znaku? Który bit (najstarszy czy najmłodszy) transmitowany jest jako pierwszy? ❓ W sprawozdaniu na przebiegach zaznacz wszystkie bity znaku oraz bit startu.
- 7.8 Powtórz punkty od 7.5 do 7.7 dla kontroli parzystości, kolejno: *Parzysty* i *Nieparzysty*. 🖱️ Pamiętaj, że jeśli ostatni bit lub bity są na niskim poziomie, to nie możesz ich wyraźnie „zobaczyć”. **UWAGA! Zmiana parametrów portu COM wymaga rozłączenia i ponownego połączenia!** Który z bitów jest bitem kontroli parzystości? Jaka jest zasada ustalania jego wartości? Czy wpływa ona na czas trwania pojedynczego bitu (przepływność)? ❓
- 7.9 Powtórz punkty od 7.5 do 7.8 dla drugiego, zadanego przez prowadzącego, znaku. 🖱️ Co uległo zmianie? Jakie elementy w porcji danych są stałe? W jaki sposób odbiornik może rozpoznać początek transmisji? ❓
- 7.10 Włącz nadawanie sekwencji pierwszego znaku dla ustawień: szybkość transmisji 9600, 8 bitów danych, BEZ parzystości, brak sterowania przepływem, 1 bit stopu.
- 7.11 Ustaw pionowe markery w sposób pozwalający określić czas trwania całej porcji danych. 🖱️ Zatrzymaj nadawanie, zaznacz opcję *Enter* i włącz nadawanie ponownie. 🖱️ Od tej chwili za pierwszym znakiem nadawany jest kod klawisza *Enter* (13_{DEC}, 001101_{BIN}). Co jest między pierwszym znakiem a *Enter*? Dlaczego do tej pory był trudny do zauważenia? ❓
- 7.12 Powtórz punkty od 7.11 do 7.11 dla 2 bitów stopu. 🖱️ Jak zmienił się przebieg? Na co wpływają bity stopu? ❓

Czytnik transponderów (kart zbliżeniowych)

- 7.13 Podłącz czytnik **HB2300** (pojedyncze gniazdo BNC z lewej strony panelu) do *Kabla portu szeregowego*, wykorzystując kabel oznaczony kolorem zielonym. Czytnik ten wysyła dane z kart (lub innych elementów) identyfikacji bezstykowej wg standardu popularnego i taniego czytnika PROX 402 firmy Corral.
- 7.14 Uruchom program ComTest. Wybierz kolejno z rozwijanych list: port **COM Kabla portu szeregowego**, szybkość transmisji (przepływność) 9600, 8 bitów danych, BEZ parzystości, brak sterowania przepływem, 1 bit stopu. Naciśnij klawisz *Połącz*. W górnym polu pojawiają się dane odbierane z czytnika. Przechwyć kilka sekwencji. Zidentyfikuj ramkę danych. 📖 Z ilu bajtów składa się ramka? Jak oznaczony jest początek ramki? Z jakiej maksymalnej odległości odczytywany jest transponder? ?

Moduł odbiorczy TRD-80 CLASSIC

- 7.15 Kanał pierwszy **CH1** przystawki oscyloskopowej **PO** podłącz do pinu **BEGIN** (górne z trzech gniazd BNC) *Modułu odbiorczego TRD-80*. Uruchom **RUN** przetwarzanie sygnału na przystawce oscyloskopowej **PO**. Wyłącz kanał drugi **CH2**. Ustaw {tryb **DSO**; **T/D** = 0,2ms; **V/D** = 1 V/dz; wyzwalenie **ON CH1**}. Ustaw poziom wyzwiania, by zapewnić odświeżanie przebiegu. Przesuń przebieg na górę ekranu, a impuls na jego lewą stronę. 📖 Jaki jest czas trwania impulsu **BEGIN**? Jaka logiczna jego wartość jest użyta do sygnalizacji początku nadawania kodu transpondera? ?
- 7.16 Włącz kanał drugi **CH2** przystawki oscyloskopowej **PO** do pinu **RS232-TTL** (gniazdo środkowe BNC). Ustaw {tryb **DSO**; **T/D** = 0,2ms; **V/D** = 1 V/dz; wyzwalenie **ON CH1**}. Ustaw poziom wyzwiania, by zapewnić odświeżanie przebiegu. 📖 O ile sygnał **BEGIN** wyprzedza początek nadawania kodu transpondera? ? Zmniejsz podstawę czasu na **T/D** = 5ms. Wykorzystując pionowe markery określ czas trwania porcji 5 bajtów (40 bitów) kodu transpondera. Jaka jest szybkość transmisji? Ile czasu trwa jeden bajt? ?
- 7.17 Zmniejsz podstawę czasu na **T/D** = 2ms. Ustaw markery na szerokość odpowiadającą czasowi trwania jednego bajta kodu. Przesuwając przebieg wyznacz wszystkie pięć bajtów kodu transpondera, zapisując za każdym razem przebieg. 📖 Określ wartość binarną każdego bajta. Jaki jest kod transpondera? ?
- 7.18 Przełącz kanał drugi **CH2** na wyjście **PWM** (dolne gniazdo BNC). Zmniejsz podstawę czasu na **T/D** = 1ms. Wiedząc, że krótszy zerowy impuls odpowiada „1”, a dłuższy „0” określ kod transpondera. Jaki jest czas trwania impulsów PWM dla „1” i „0”? Jaki jest kod transpondera? ?

8. Wykonanie sprawozdania

Nie należy umieszczać w sprawozdaniu podstaw teoretycznych, ani opisów stanowiska laboratoryjnego. Sprawozdanie musi zawierać wszystkie wyniki pomiarów i obserwacji prezentowane wg kolejności ich wykonania. Każdy z nich musi być opatrzony numerem punktu instrukcji wg, którego został zarejestrowany. W sprawozdaniu muszą się znaleźć odpowiedzi na wszystkie postawione w instrukcji pytania oraz odpowiedni komentarz do uzyskanych wyników badań symulacyjnych.

Zarówno opisy, jak i odpowiedzi, mają być zwięzłe, ale przedstawione pełnymi zdaniami. Wnioski powinny zawierać podsumowanie przeprowadzonych badań. Szczególny nacisk należy położyć na zaprezentowanie różnic oraz podobieństw pomiędzy poszczególnymi wynikami i obserwacjami, np. różnice i podobieństwa w formatach transmisji, czy zależności czasowe przy zmianach szybkości transmisji.

9. Literatura

Kubas M., Molski M., *Karta elektroniczna bezpieczny nośnik informacji*, Warszawa, MIKOM 2002

Mielczarek W., *Szeregowe interfejsy cyfrowe*, Gliwice, HELION 1993

Bogusz J., *Lokalne interfejsy cyfrowe w systemach cyfrowych*, Warszawa, BTC 2004

Strona internetowa: <http://www.arczer.neostrada.pl/komputery.htm>

Strona internetowa: <http://forbot.pl/blog/>