Karty RFID w praktyce

Autor: Piotr Rzeszut

Artykuł na konkurs wakacyjny

FORBOT.PL
ROBOTYKA AMATORSKA

Wstęp

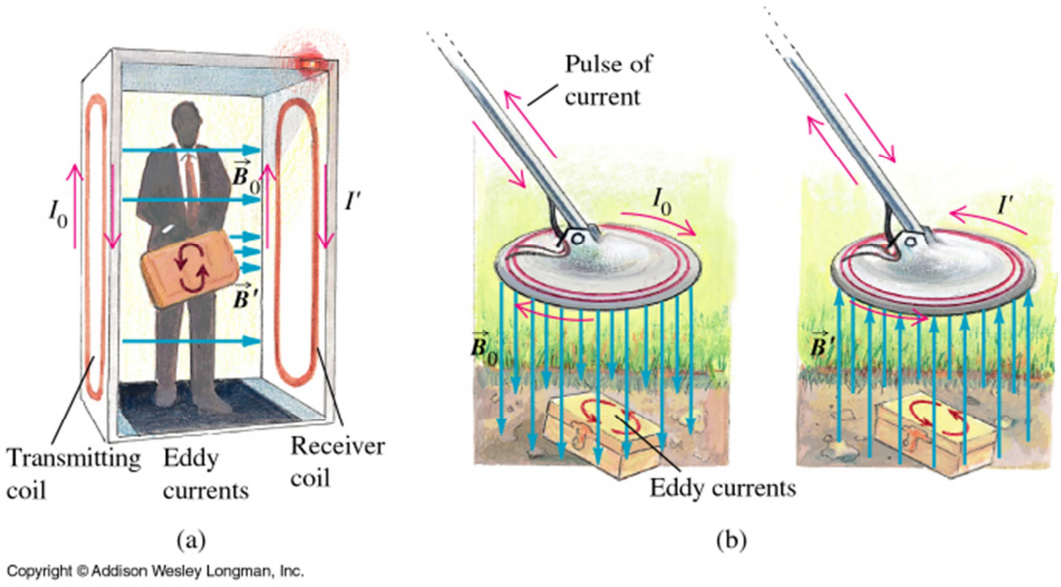
W dzisiejszych czasach nikt z nas nie wyobraża sobie życia bez zamków i kluczy, a zakupów w supermarkecie bez kodów kreskowych. Co jednak łączy te dwa tematy? I gdzie ich powiązanie z elektroniką a konkretniej robotyką? Odpowiedź na pierwsze pytanie będzie jednocześnie tematem całych naszych wspólnych rozważań, a mianowicie brzmi ona: RFID. Co jednak kryje się pod tymi czterema tajemniczymi literami? Z języka Angielskiego to skrót ***R**adio **F**requency **ID**entifiaction*, czyli w wolnym tłumaczeniu „identyfikacja za pomocą fal radiowych". Już chyba teraz każdy przypomniał sobie o „białych magicznych kartach" otwierających drzwi po zbliżeniu do czytnika czy nowoczesnych zbliżeniowych kartach płatniczych, albo karnetach na stokach, basenach czy biletach okresowych w komunikacji miejskiej, chipach identyfikacyjnych dla zwierząt, czy tych wszczepianych ludziom jako „karty płatnicze". Niektórzy z Was może słusznie wskażą na systemy zabezpieczające przed kradzieżą jako kolejne przykłady wykorzystania systemów RFID, a dociekliwsi wspomną o eksperymentalnych „metkach", które mogłyby być „kasowane" poprzez przejazd wózka przez odpowiednią bramkę… W moim artykule postaram się więc przybliżyć tę z jednej strony nie najnowszą, ale z drugiej strony bardzo szybko rozwijającą się i przyszłościową technologię, opisać zasadę działania takich kart, a także przedstawić po krótce szeroki wachlarz ich zastosowania i zaproponować nowe konkurencje dla robotów z ich zastosowaniem.

Wszystkie przedstawione w niniejszym artykule informacje mają jedynie charakter informacyjny i powstały w celach naukowych. AUTOR NIE PONOSI ODPOWIEDZIALNOŚCI ZA ICH WYKORZYSTANIE a także przypomina iż wykorzystanie niektórych przedstawionych tu informacji poza laboratorium czy areną zawodów związanych z robotyką może być niezgodne z prawem.

Szczypta historii i trochę teorii

Technologie RFID wywodzą się od bardzo prostego i dobrze znanego zagadnienia wykrywania metali, nad którym prace prowadzono od lat 40-stych XX w. Warto wspomnieć że jeden z pierwszych działających wykrywaczy metali został zbudowany w czasach II Wojny Światowej przez dwóch polskich poruczników.

Zasada działania takiego wykrywacza jest w gruncie rzeczy niezmiernie prosta – przepływ prądu zmiennego przez cewkę powoduje powstanie wokół niej zmiennego pola magnetycznego, to zaś z kolei, jak wiemy z lekcji fizyki, powoduje indukcję prądów wirowych w umieszczonych w polu metalowych przedmiotach. Skoro w nich też zaczyna płynąć prąd, to także powoduje on postawnie wokół przedmiotu pola magnetycznego, które, zgodnie z regułami fizyki, osłabia pole w cewce i wpływa na przepływ prądu przez nią. Wykrywanie tych zmian pozwala nam w dużym uproszczeniu stwierdzić czy w polu magnetycznym wytwarzanym przez cewkę znajduje się jakiś kawałek metalu czy nie.



Prądy wirowe indukowane w elementach metalowych umieszczonych w zmiennym polu magnetycznym umożliwiają ich wykrycie za pomocą wykrywaczy metali.
(źródło: http://www.physics.sjsu.edu/becker/physics51/induction.htm)

Stąd już niedaleka droga do znanych każdemu bramek antykradzieżowych obecnych w każdym supermarkecie. Jedyną różnicą zasady ich działania w stosunku do zwykłego wykrywacza matali jest to, iż częstotliwość z jaką zmienia się pole magnetyczne jest ściśle dopasowania do częstotliwości rezonansowej cewki (a dokładniej ujmując układu cewka-kondensator) umieszczonej w nalepce (czy ogólnie mówiąc dowolnym tagu), w taki sposób, że wykrywane są tylko zmiany pola powstałe w wyniku umieszczenia w nim (nie zdezaktywowanego w kasie) tagu – inne elementy metalowe nie są wykrywane.

Nalepka zabezpieczająca zawiera cewkę, która pozwala na jej wykrycie przy wykorzystaniu ściśle określonej częstotliwości zmian pola magnetycznego.
(źródło: http://en.wikipedia.org)

A teraz wprowadźmy jeszcze jedną małą modyfikację – niech nasza „metka” nie emituje jedynie wtórnej „fali”, ale niech także, dzięki układowi scalonemu, odpowiednio moduluje (zmienia parametry) tą falę i w ten sposób (skoro już byliśmy wstanie odróżnić obecność metalu do jej braku, a

potem wykryć tylko konkretną cewkę) przesłać z naszej metki informacje do anten. Tym sposobem doszliśmy do istoty działania systemów RFID, więc pora na poznanie konkretów.

Czas na praktykę

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 przedstawię 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)

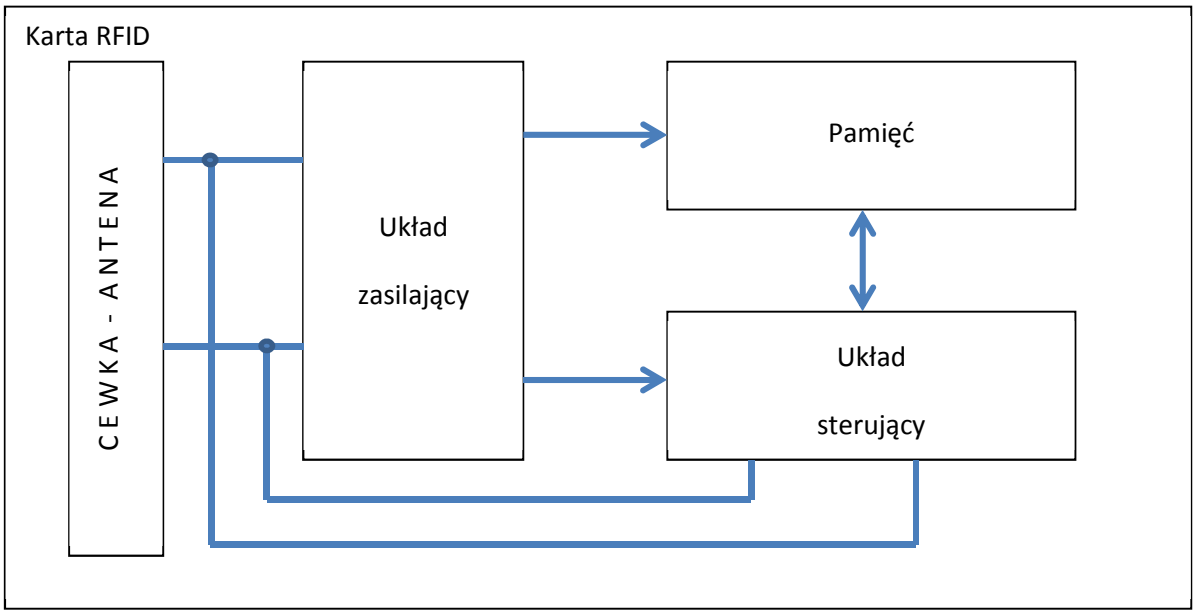
Mifare

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.

W moim artykule nie mógłbym opisać wszystkich standardów kart zbliżeniowych, gdyż prawdopodobnie wyczerpujący opis samego tylko standardu Mifare w wybranych jego odmianach stworzyłby książkę, dlatego zdecydowałem się na przybliżenie Wam najpopularniejszego, najprostszego i chyba najtańszego (choć nie pozbawionego wad, o czym w dalszej części artykułu) standardu – RFID UNIQUE 125kHz.

Budowa karty RFID

Karta, czy dowolny inny tag RFID składa się, z punktu widzenia laika, z 2 ważnych elementów: cewki (pełniącej jednocześnie rolę elementu zasilającego układ oraz nadawczo-odbiorczego) oraz układu scalonego odpowiedzialnego za funkcjonowanie karty. 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):



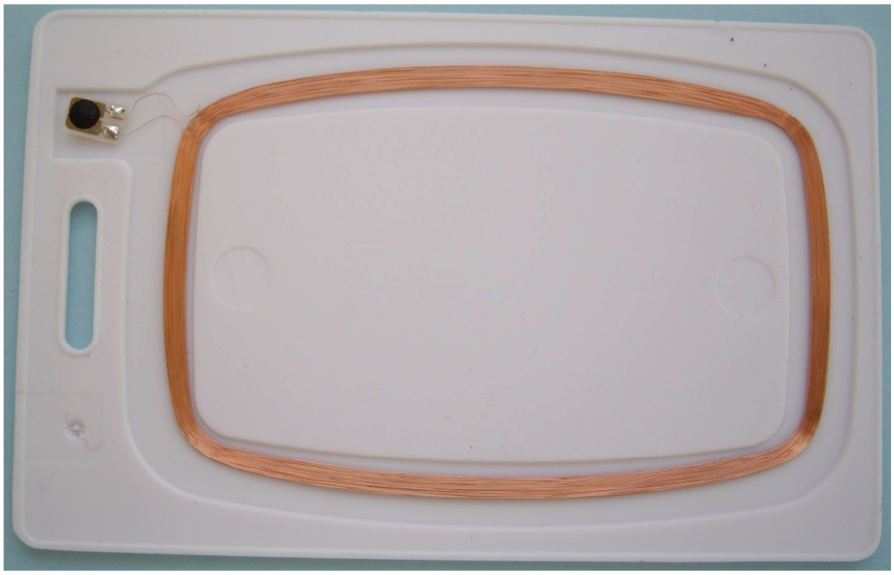
Zasada działania układu jest prosta i zgodna z tym co wcześniej stwierdziliśmy:

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.

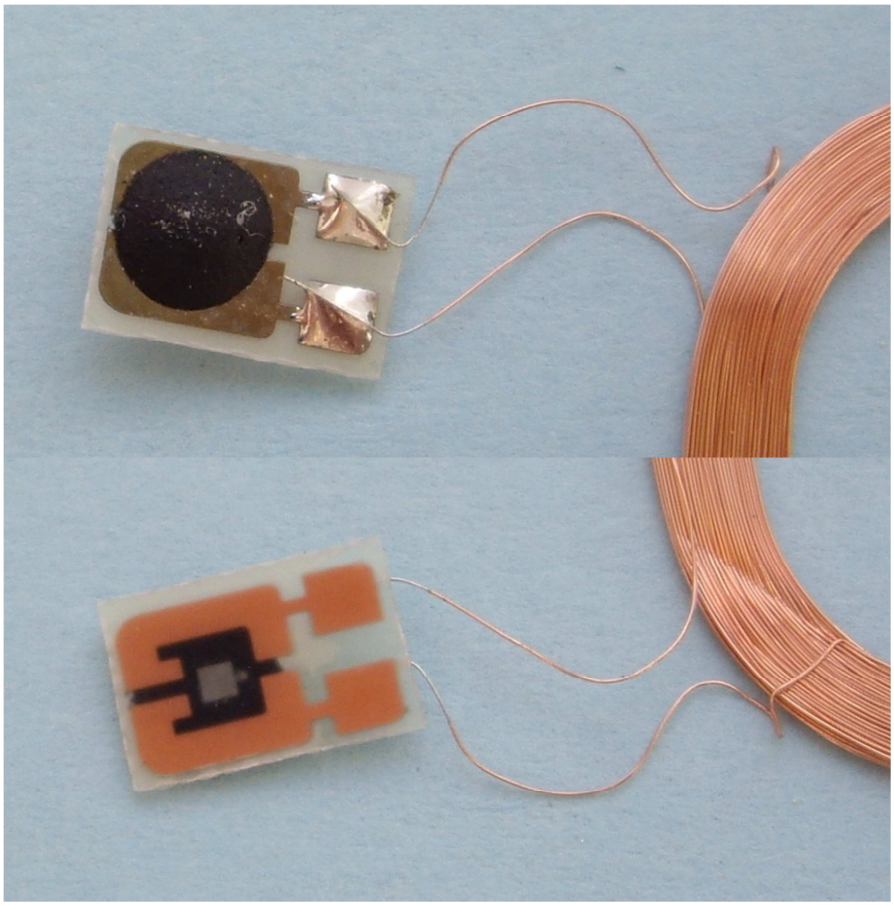
W omawianych przez nas szczegółowo kartach UNIQUE układ sterujący na podstawie sygnału częstotliwości nośnej generuje sygnał taktujący nadawanie, a poprzez załączanie lub odłączanie od obwodu cewki dodatkowej rezystancji zmienia natężenie prądu przez nią płynącego, a co za tym idzie zmienia natężenie 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 sadek 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 (na przykładzie karty RFID UNIQUE 125kHz w obudowie ClamShell):

Przednia strona obudowy karty RFID



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

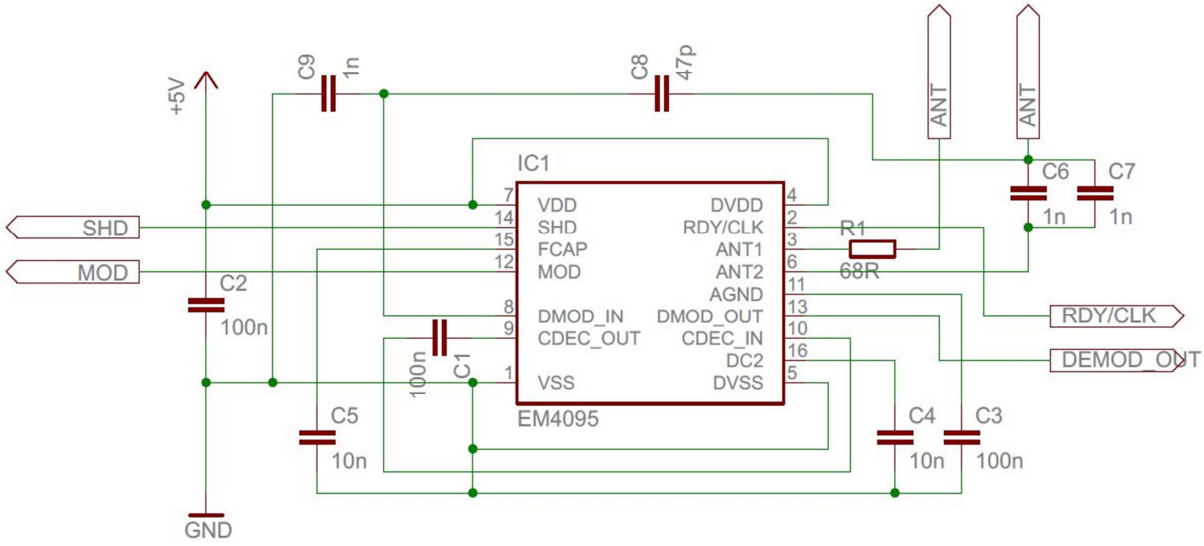


Układ scalony w powiększeniu – na dolnym zdjęciu (widok od dołu) widoczne są płytka krzemowa, na której wykonany jest układ scalony (szary prostokąt) oraz kondensator wykonany w formie fragmentów miedzi na laminacie (złote kształty przypominające litery C w odbiciu lustrzanym); na górnym zdjęciu widoczna kropla tworzywa sztucznego chroniąca delikatną strukturę układu scalonego od strony jego montażu.

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.

Ja jednak chciałbym Wam przedstawić odczyt danych z karty na nieco niższym poziomie. Użyjemy jedynie układu generującego częstotliwość nośną i demodulującego sygnał (zmiany amplitudy, o których pisałem) do postaci cyfrowej. W tej kategorii „czytników”, a raczej demodulatorów RFID UNIQUE bardzo prostym rozwiązaniem jest zastosowanie układu scalonego EM4095 w następującej aplikacji:

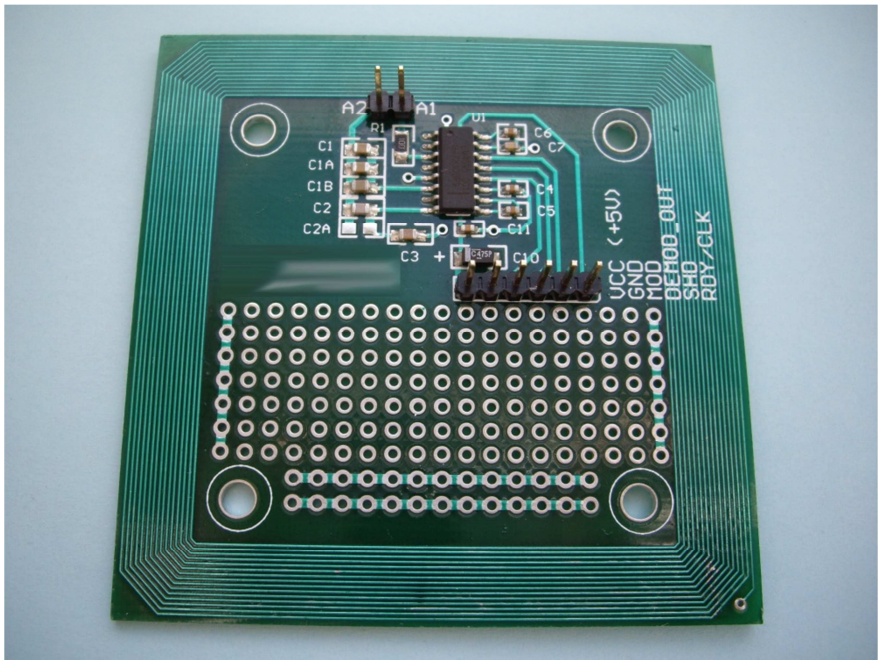


Schemat prostego układu demodulatora RFID w oparciu o układ EM4095

Opis wyprowadzeń:

1. ANT – złącza anteny. Podłączamy do nich cewkę o indukcyjności ok. 736 µH lub posiadającą 32 zwoje i średnicę 8 cm
2. RDY/CLK – złącze na którym dostępny jest w standardzie TTL sygnał nadawany przez antenę (jest to przebieg prostokątny) – do naszych celów pozostawiamy niepodłączony, choć możemy go wykorzystać jako generator odniesienia czasowego przy dekodowaniu sygnału (jak pamiętacie może nadanie jednego bitu zajmuje 64 cykle, których trwanie możemy zliczać z wykorzystaniem tego wyjścia)
3. DEMOD_OUT – zdemodulowany sygnał odebrany z karty – to wyjście będziemy wykorzystywać do dekodowania jej numeru
4. MOD – wejście układu EM4095, które umożliwia modulację sygnału nadawanego przez antenę w celu zapisu specjalnych kart. Jako że nie będziemy wykorzystywać tej funkcji podłączamy na stałe to wejście z masą (GND)
5. SHD – złącze pozwalające na włączenie lub wyłączenie układu i jego reset. Aby zresetować układ należy na tym złączu wygenerować zbocze opadające i utrzymać stan niski.

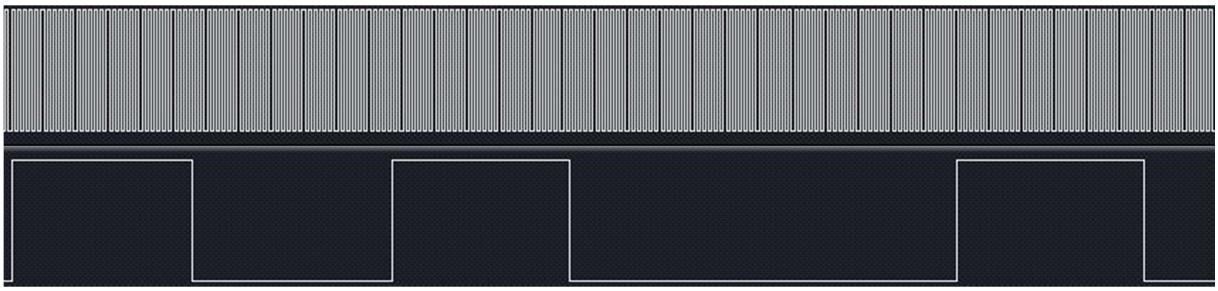
Możemy także nabyć gotowy i przetestowany moduł z tym układem i anteną wykonaną jako ścieżka na płytce PCB:



Odczyt i dekodowanie

Kiedy nasz hardware uzupełnimy o niemalże dowolny procesor AVR (ja do testów stosowałem procesor ATMega328p) możemy przystąpić do sedna sprawy – odczytu i dekodowania informacji odebranych z karty. Ale nie od razu Kraków zbudowano, dlatego musimy do sprawy podejść krok po kroku, zdobywając za każdym krokiem nieco więcej informacji.

Na początek przyjrzyjmy się przebiegom z wyjść CLK i DEMOD_OUT:

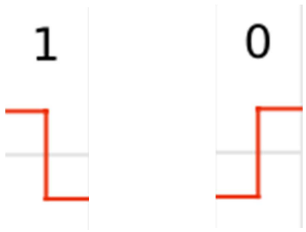


Wynik próbkowania linii CLK i DEMOD_OUT za pomocą analizatora stanów logicznych.

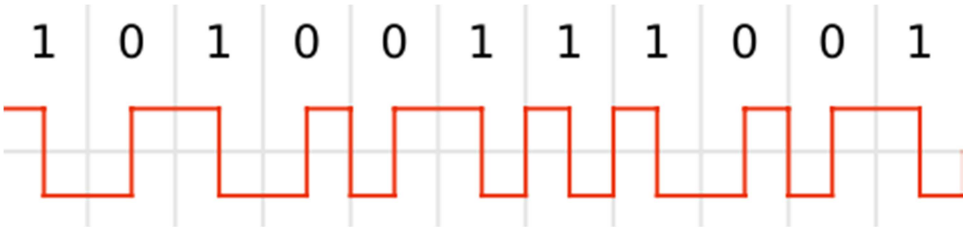
Na początku jedyne co możemy stwierdzić to to, że sygnał zegarowy jest przebiegiem prostokątnym o częstotliwości zbliżonej do 125kHz, a długość impulsów wynosi 32 lub 64 cykle zegarowe (jeśli nie wierzycie na słowo – policzcie). Teraz należy sięgnąć do noty katalogowej naszej karty (a konkretnie zawartego w niej układu EM4100), gdzie czytamy (po angielsku rzecz jasna), że dane są kodowane za pomocą kodu Manchester…

Kod Manchester

Kod Manchester jest w rzeczywistości bardzo prostym kodem. Otóż, 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:

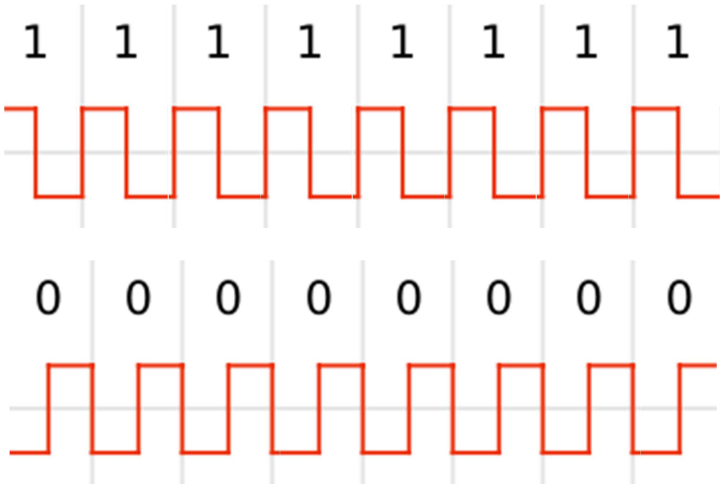


Jeżeli teraz spojrzymy na większy fragment danych nadawanych z wykorzystaniem tego kodu to zauważymy jego bardzo ciekawą właściwość, pomagającą w jego zdekodowaniu:

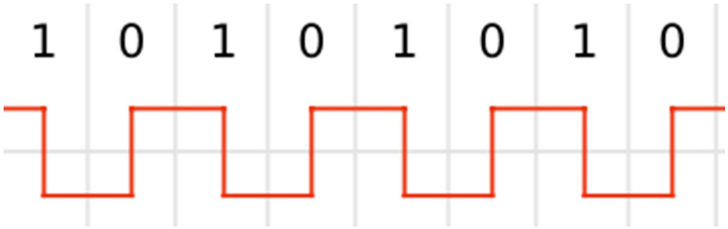


(źródło obrazków: http://en.wikipedia.org)

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 dziecinnie proste. Na początku przyjmiemy jakąś wartość nadawanego bitu, np. B=1 (niestety karty, w przeciwieństwie do np. pilotów RC5, nadają dane ciągle, „w kółko", nie stosując żadnej zależności czasowej dla wskazania początku nadawania pakietu danych) i postępujmy 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 (na razie ręcznie na ekranie komputera) możemy zdekodować całą ramkę… Ale właśnie… gdzie właściwie zaczyna się i kończy ta ramka, oraz co oznacza ten bezładny ciąg zer i jedynek?

Ramka RFID

Teraz przyszła kolej na poznanie struktury danych przesyłanych przez naszą kartę. Najprościej, podążając za autorem oryginalnych dokumentacji najprościej przedstawić ramkę w następującej formie:

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 (*even parity*) dla każdego z wierszy tabelki. Jeśli w danym wierszu znajduje się nieparzysta ilość jedynek to bit przyjmuje wartość 1, w przeciwnym wypadku przyjmuje on wartość 0. CP0…CP3 – bity parzystości (*even parity*) dla każdej z kolumn tabelki. Działają na takiej samej zasadzie jak parzystość wierszy.
				D04	D05	D06	D07	P1	
32 bity – unikalny kod karty				D08	D09	D10	D11	P2	
				D12	D13	D14	D15	P3	
				D16	D17	D18	D19	P4	
				D20	D21	D22	D23	P5	
				D24	D25	D26	D27	P6	
				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

Niestety, jak już wcześniej wspomniałem, karty nadają swoje dane w pętli nie stosując żadnego czasowego wyróżnika dla sygnalizowania rozpoczęcia transmisji danych (jak to ma miejsce np. w standardzie RC5). 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. 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 (jeśli nie wierzycie to sprawdźcie sami). 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.

To jednak nie wszystko. Z powodu braku czasowego oznaczenia początku ramki nasze oprogramowanie (które za niedługo będziemy tworzyć) 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 bitu parzystości jest różna od wyliczonej na podstawie danych).

Zatem czas napisać prosty algorytm dekodowania danych:

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.

I tym oto sposobem… Na razie w teorii… Odczytaliśmy dane z karty i możemy je wykorzystać do identyfikacji osobnika jej używającego. Teraz czas na napisanie właściwego programu na procesor AVR (w przykładzie ATMega328P) wykonującego wszystkie te operacje w mgnieniu oka.

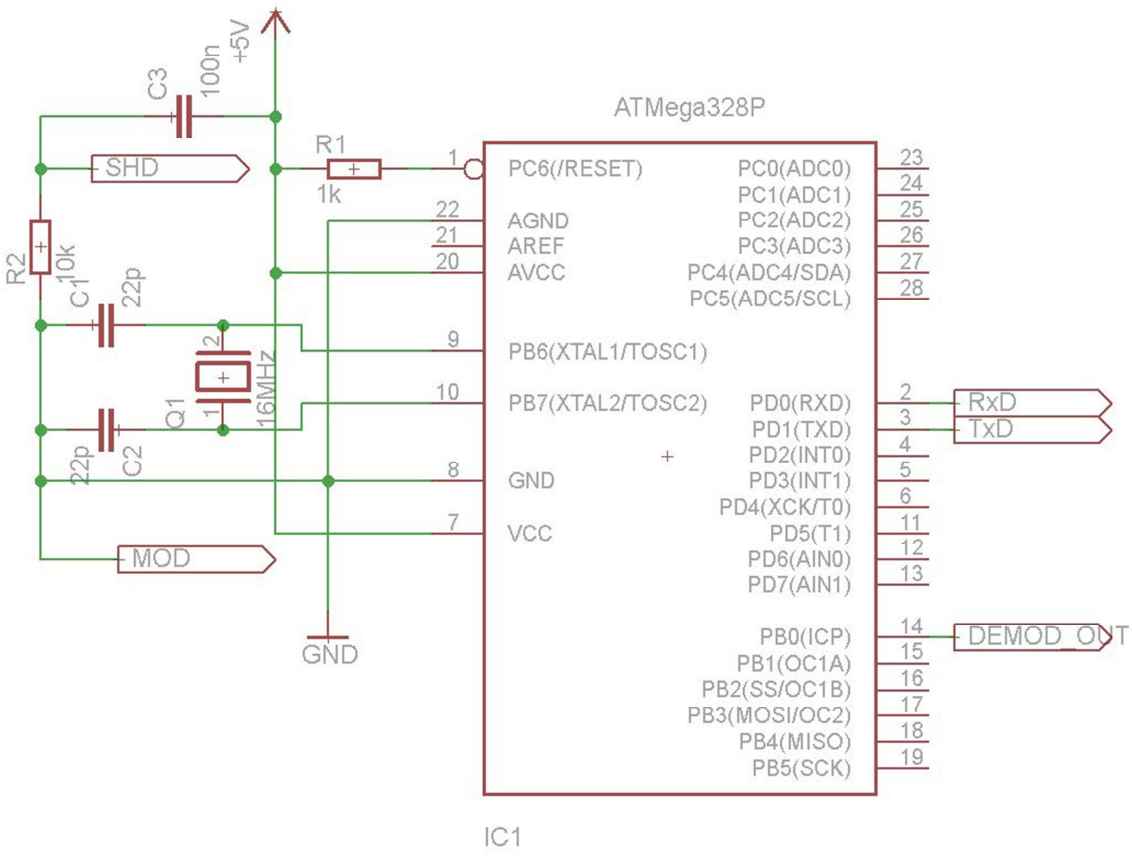
Program odczytujący karty

O ile zapisanie większości omówionych przez nas kroków w formie programu nie powinno stanowić większego problemu (w gruncie rzeczy wszystko sprowadza się do kilku pętli, przesunięć bitowych, paru instrukcji warunkowych i voila), o tyle ciekawe jest zagadnienie pomiaru czasu trwania impulsu. Rzecz jasna moglibyśmy zaangażować do tego cały czas pracy procesora i oczekiwać na kolejne impulsy w pętli i po ich wykryciu zapisywać różnice czasów odmierzone za pomocą timera, jednak całe te działania może za nas wykonać sam timer.

Jeśli ustawimy Timer1 w tryb ICR to wtedy jeśli na odpowiednim wejściu pojawi się zbocze wybrane za pomocą bitu ICES1(opadające lub narastające) wykonane zostanie przerwanie i dodatkowo natychmiast aktualna wartość timera zostanie zapisana do specjalnego rejestru ICR1. Teraz wystarczy tylko od tej wartości odjąć poprzednio zapamiętaną w jakiejś zmiennej, aby obliczyć szerokość impulsu i szybko zmienić wartość bitu ICES1, aby kolejne przerwanie zostało wywołane przez przeciwne zbocze. Tym sposobem, jeśli dodatkowo w przerwaniu będziemy wykrywać odebranie całej potencjalnej paczki 64 bitów i wtedy dokonywać jej dekodowania, cała praca dekodera będzie działała niezależnie od programu głównego i będzie praktycznie niezauważalna – po prostu jeśli ktoś zbliży kartę to w odpowiedniej zmiennej znajdzie się jej numer i zostanie dodatkowo ustawiona flaga informująca o tym, że odebrano jakieś ciekawe informacje.

Kompletne rozwiązanie dla procesora ATMega328p dekodujące i wysyłające poprzez UART (9600bd) numer zbliżonej karty znajdziecie w załącznikach do tego artykułu.

Aaaaa….. Prawie bym zapomniał o schemacie połączeń do procesora:



Bezpieczeństwo kart UNIQUE 125kHz

Bezpieczeństwo w teorii...

W teorii karty standardu UNIQUE są dosyć bezpieczne – numery powinny być niepowtarzalne, a karty równie bezpieczne jak klucze. Czy jednak rzeczywiście tak jest?

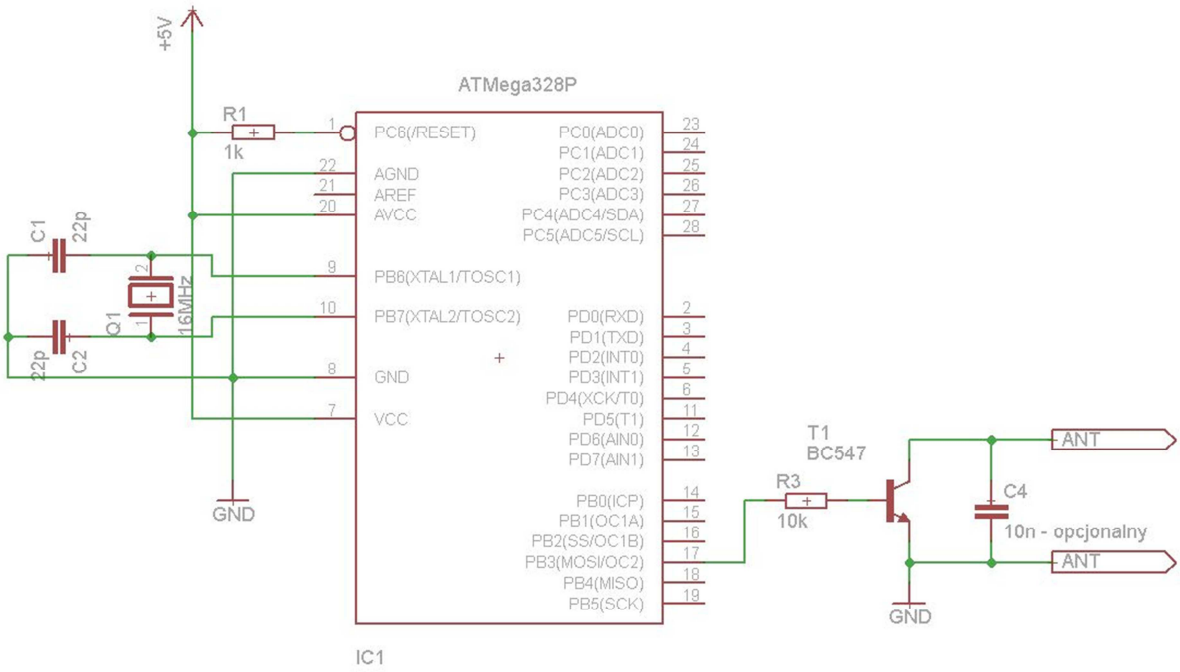
...i praktyce

Niestety nie. 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.

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. Najciekawszy z odnalezionych przeze mnie 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.

Emulujemy kartę RFID.

Jeśli pamiętacie co mówiłem o zasadzie działania oryginalnych kart, to przypomnicie sobie szybko, ze modulacja sygnału polega na zmianie natężenia prądu płynącego przez cewkę w karcie. Najprościej możemy w praktyce tego dokonać po prostu zamykając i otwierając obwód za pomocą choćby tranzystora NPN. W związku z tym przykładowy schemat emulatora mógłby wyglądać tak:



Do złącz oznaczonych ANT podpinamy antenę. Może być to taka sama antena, jaką wykorzystywaliśmy do budowy odbiornika, choć równie dobrze możemy takową pozyskać z demontażu karty. W zależności od zastosowanej cewki montaż kondensatora C4 może okazać się zbędny, a ponadto w niektórych przypadkach może zmieniać na tyle parametry obwodu LC, że uniemożliwi poprawną pracę układu. Jeśli nie mamy czasu na obliczenia najlepiej sprawę tę rozstrzygnąć wykonując odpowiednie testy.

Oprogramowanie

Tym razem oprogramowanie na początku musi wykonać operacje odwrotne do tych wykonywanych podczas dekodowania numeru karty, a potem „w kółko” nadawać wygenerowany ciąg bitów. O ile kodowanie jest kwestią bardzo prostą – wystarczy tylko umieścić odpowiednie elementy stałe (bity startu i stopu), dane i wyliczone bity parzystości na odpowiednich miejscach – o tyle samo nadawanie tak wygenerowanego ciągu bitów jest warte omówienia. Algorytm ten wygląda następująco (jedyne co musimy w nim zapamiętywać to ostatnio nadawany bit i ilość nadanych zboczy):

1. Jeśli nadaję parzyste zbocze to idź do kroku 2, w przeciwnym wypadku idź do kroku 6
2. Jeśli aktualnie nadawany bit jest taki sam jak poprzednio nadawany to zmień stan wyprowadzenia nadawczego na przeciwny, w przeciwnym wypadku nie rób nic.
3. Zapamiętaj aktualnie nadawany bit jako ostatnio nadawany bit
4. Przygotuj do nadawania kolejny bit z wygenerowanej ramki
5. Idź do kroku 7
6. Zmień stan wyprowadzenia nadawczego na przeciwny
7. Zwiększ licznik zbocz

8. Jeśli licznik zbocz osiągnie wartość 127 to zresetuj go do wartości 0 (nadaliśmy całą ramkę)
9. Wróć do kroku 1 po upływie odpowiedniego czasu (połowy jednego bitu – 32 cyklach częstotliwości nośnej)

I w tym wypadku najlepiej do odmierzania czasu posłużyć się odpowiednio skonfigurowanym timerem (w trybie CTC), dzięki czemu i w tym wypadku całe nadawanie odbywa się prawie niezauważalnie z punktu widzenia programu głównego.

Podsumowanie – a co z robotyką?

Jak na razie dowiedzieliśmy się co-nieco o teorii działania kart zbliżeniowych, różnych ich modelach oraz w miarę szczegółowo poznaliśmy standard RFID Unique 125kHz. Umiemy za pomocą mikrokontrolera AVR i układu EM4095 zdekodować dane z karty, a nawet symulować jej działanie za pomocą drugiego mikrokontrolera. Co jednak ma do tego robotyka?

Oczywiście chciałbym przedstawić parę pomysłów (oprócz wątpliwego zabezpieczenia swojego laboratorium i warsztatu) na nowe konkurencje dla robotów:

1. Bieg na orientację – jazda na precyzję
   Roboty poruszają się po planszy pod którą umieszczone są karty RFID. Karty stanowią wskazówki dla robotów jak dotrzeć do następnej karty lub mety. W konkurencji liczy się precyzja wykonywanych skrętów i pokonywanych odległości oraz czas przejazdu. W kwestii technicznej można by wykorzystać :
   a) wykonane na bazie procesorów ATTiny85 emulatory kart lub sieć aktywnych emulatorów kart (można sterować kilkoma „kartami" z jednego procesora delikatnie modyfikując kod emulatora zaprezentowany przeze mnie), wtedy określone bajty numeru ID informowałyby o kierunku i odległości do następnego punktu
   b) zwykłe karty – ich numery byłyby spisywane przez organizatorów i każdy z robotów otrzymywałby kartę SD w której zapisane byłyby numery kart oraz odpowiadające im odległość i skręt.

   Zwyciężałby robot, który w najkrótszym czasie zakończy swój przejazd zatrzymując się możliwie blisko wyznaczonego punktu końcowego.

2. Zbieracz kart
   Konkurencja polegałaby (podobnie jak zbieranie kolorowych dysków) na zbieraniu kart o określonych numerach (także podawanych na kartach SD lub karty byłyby emulatorami o z góry określonych kodach). Zebranie karty przeciwnika byłby karane punktami ujemnymi, a dozwolone techniki mogłyby obejmować np. zakłócanie pracy odbiornika przeciwnika.