

Politechnika Gdańska



**Wydział Elektroniki, Telekomunikacji
i Informatyki**

Katedra:	Systemów Automatyki
Imię i nazwisko dyplomanta:	Szymon Czaja
Forma i poziom studiów:	dzienne inżynierskie
Kierunek studiów:	Automatyka i Robotyka

PRACA DYPLMOWA INŻYNIERSKA

Temat pracy: Układ sterujący bezprzewodowo modułem RCX zestawu LEGO Robotic Invention System 2.0, wykorzystujący interfejs Bluetooth.

Opiekun pracy: dr inż. Krzysztof Cisowski

Gdańsk, 2010 rok

OŚWIADCZENIE

Oświadczam, że niniejszą pracę dyplomową wykonałem samodzielnie. Wszystkie informacje umieszczone w pracy uzyskane ze źródeł pisanych oraz informacje ustne pochodzące od innych osób zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami.

.....

Spis rysunków :

- Rys. 1: Moduł LEGO RCX 2.0.
- Rys. 2: Czujnik światła wchodzący w skład zestawu [1].
- Rys. 3: Czujnik dotyku wchodzący w skład zestawu [1].
- Rys. 4: Silnik wchodzący w skład zestawu [1].
- Rys. 5: Wieża nadawczo - odbiorcza wchodząca w skład zestawu [6].
- Rys. 6: Przykładowy fragment kodu w języku RCX CODE [3].
- Rys. 7: Klocki funkcyjne dostępne w RCX CODE [3].
- Rys. 8: Okno programu Nqcedit wraz z fragmentem kodu w języku NQC [5].
- Rys. 9: Struktura sieci Bluetooth oparta na pikosieciach [4].
- Rys. 10: Organizacja ramki w sieci Bluetooth [4].
- Rys. 11: Mikroprocesor ATmega8 wraz z opisem wyprowadzeń [7].
- Rys. 12: Moduł Bluetooth BTM-222 [8].
- Rys. 13: Filtr RC służący do prostowania sygnału PWM [10].
- Rys. 14: Schemat zasilania układu.
- Rys. 15: Schemat połączenia mikrokontrolera z modulem Bluetooth.
- Rys. 16: Schemat połączenia mikrokontrolera z filtrem RC.
- Rys. 17: Projekt płytki drukowanej.
- Rys. 18: Rozkład ścieżek i otworów pod elementy.
- Rys. 19: Okno startowe programu na PC.
- Rys. 20: Okno konfiguracyjne programu.

Spis treści:

1. Cel i zakres pracy.....	5
2. Zestaw LEGO Robotic Invention System, oraz moduł RCX 2.0 i jego możliwości.....	6
3. Komunikacja bezprzewodowa oparta na sieci Bluetooth.....	11
4. Projekt urządzenia do komunikacji bezprzewodowej opartej na interfejsie Bluetooth.....	13
4.1 Idea działania urządzenia.....	14
4.2 Wykorzystywane elementy.....	15
4.2.1 Mikroprocesor Atmega8L.....	15
4.2.2 Moduł Bluetooth BTM-222.....	18
4.3 Interfejs komunikacyjny pomiędzy mikroprocesorem a modulem RCX 2.0.....	20
4.4 Schemat połączeń elektrycznych.....	21
4.5 Projekt płytki drukowanej.....	24
5. Oprogramowanie komputera PC.....	25
6. Podsumowanie.....	29
7. Literatura.....	30

1. Cel i zakres pracy.

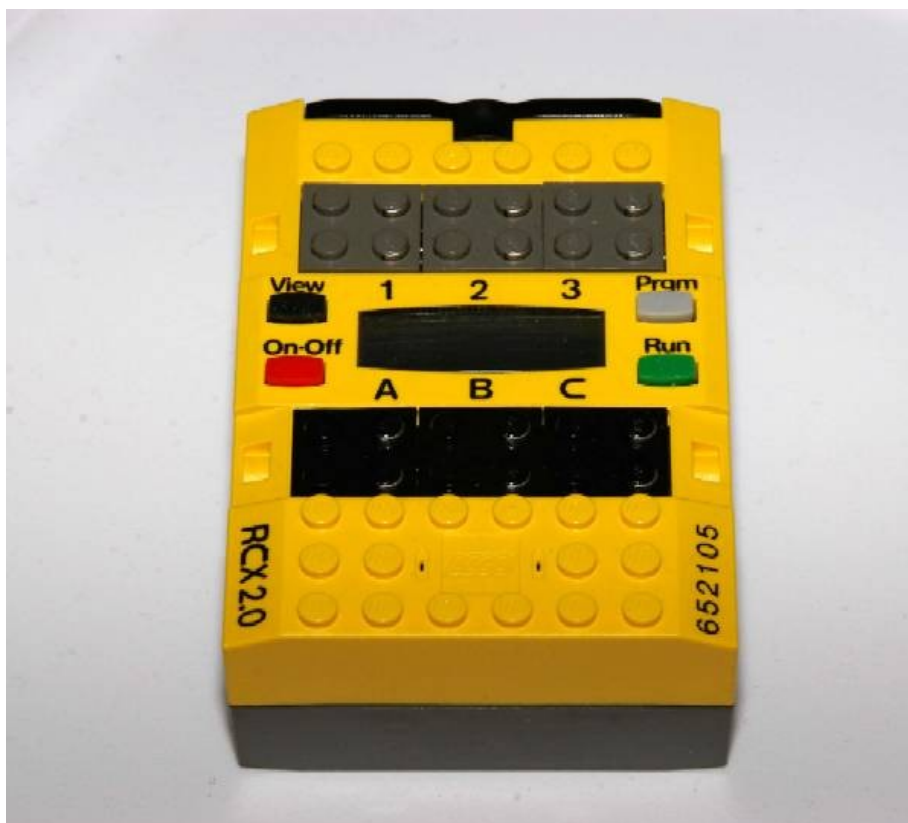
Celem pracy było zaprojektowanie i wykonanie urządzenia umożliwiającego bezprzewodową komunikację komputera PC z jednostką centralną RCX 2.0 zestawu LEGO Robotic Invention System, umożliwiającego sterowanie zestawem w czasie rzeczywistym za pośrednictwem komputera PC. Zakres pracy obejmuje również napisanie oprogramowania zapewniającego prawidłową pracę urządzenia, oraz oprogramowania umożliwiającego obsługę tego urządzenia (oprogramowanie na komputer PC). Aby zrealizować podane cele, konieczne było odpowiednie dobranie elementów elektronicznych, odpowiednie połączenie ich oraz zaprojektowanie płytki drukowanej odpowiadającej schematowi połączeń elementów.

2. Zestaw LEGO Robotic Invention System, oraz moduł RCX 2.0 i jego możliwości.

Omawiany zestaw nie doczekał się zbyt wielu publikacji omawiających go, stąd większość informacji pochodzi wprost ze strony internetowej producenta lub stron internetowych fanów tego zestawu. Adresy tych stron zostały zamieszczone w bibliografii. Zestaw LEGO Robotic Invention System składa się z ponad 700 elementów co czyni go użytecznym w wielu zastosowaniach. Jednak najbardziej znaczące to zastosowanie w dydaktyce. Zestaw ten wyposażony jest (oprócz zwykłych klocków LEGO) w takie elementy jak:

- 2 silniki 9V
- 1 czujnik światła
- 2 czujniki dotykowe
- programowalny klocek „BRICK” RCX 2.0

Wyżej wymienione elementy wchodzi w skład każdego zestawu Lego Robotic Invention System. Oprócz nich, dostępne są również inne udostępnione przez producenta czujniki, jak np. czujnik rotacyjny rozróżniający 16 pozycji na obrót oraz czujnik temperatury mierzący temperaturę w zakresie od -20°C do $+50^{\circ}\text{C}$, czujniki firmy Pitsco współpracującej z firmą LEGO (czujniki wilgotności, ruchu, ciśnienia, dźwięku), jak również różnorodne czujniki opracowane przez fanów tej gałęzi działalności firmy LEGO.



Rys. 1: Moduł LEGO RCX 2.0.

Czujnik światła wchodzący w skład zestawu jest czujnikiem aktywnym, składającym się głównie z diody oraz fototranzystora. Zadaniem diody jest oświetlenie otoczenia czujnika, natomiast fototranzystor dokonuje pomiaru natężenia światła docierającego do niego. Fototranzystor ten jest wrażliwy głównie na podczerwień, co czyni go przydatnym również do m.in. lokalizacji innych pracujących w otoczeniu modułów RCX.



Rys.2: Czujnik światła wchodzący w skład zestawu [1].

Czujnik dotykowy, jest natomiast czujnikiem biernym, zrealizowanym jako zwykły przełącznik analogowy. Zwarty posiada bardzo niewielką rezystancję, rozarty natomiast – nieskończoną rezystancję. Taka organizacja czujnika pozwala na podłączenie kilku czujników do jednego wejścia RCX, co jest istotne przy rozbudowanych projektach ze względu na istnienie tylko 3 wejść w module.



Rys. 3: Czujnik dotyku wchodzący w skład zestawu [1].

Silniki wchodzące w skład zestawu są silnikami prądu stałego o nominalnym napięciu zasilania równym 9V i sprawności 80%. Silniki posiadają wbudowaną przekładnię zwiększającą moment napędowy silnika kosztem prędkości obrotowej. Nie jest to jednak problem, gdyż w skład zestawu wchodzi wiele różnorodnych elementów pozwalających na konstrukcje nawet rozbudowanych przekładni. Silniki sterowane są przez moduł RCX za pomocą prostokątnego sygnału PWM (Pulse Wide Modulation). Pobór prądu nieobciążonego, obracającego się silnika to 5-10 mA, a prędkość obrotowa wynosi ok.350 obrotów/minute. Prąd pobierany przez silnik zablokowany wzrasta do 250-350 mA.



Rys. 4: Silnik wchodzący w skład zestawu [1].

Najbardziej złożonym elementem całego zestawu jest jednostka sterująca „BRICK” RCX 2.0.

Odpowiada ona m.in. za:

- Sterowanie silnikami, w tym modulacja PWM
- Zarządzanie konwersją A/D sygnałów z wejść RCX
- Komunikację z użytkownikiem (obsługa wyświetlacza i głośniczka)
- Obsługa komunikacji z wieżą IR podłączoną do komputera PC za pośrednictwem której możliwe jest programowanie zestawu

Wszystkie instrukcje służące do obsługi m.in. wyżej wymienionych operacji znajdują się w 16KB pamięci ROM. Należy też wspomnieć, że wejścia modułu RCX nie są wejściami tylko biernymi. Ponieważ czujniki wchodzące w skład zestawu nie posiadają zasilania, nie są one w stanie same generować sygnałów napięciowych. Z tego powodu wyjścia modułu (jeżeli pracują w trybie aktywnym) muszą jednocześnie zasilać czujniki jak i odbierać informacje od nich pochodzące. Rozwiązanie tego problemu wygląda następująco: najpierw, przez 150us, na wejście wystawiane jest napięcie 8V, następnie przez 32 us napięcie jest odłączane i dokonywany jest pomiar napięcia tak jakby wejście pracowało w trybie biernym.

Komunikacja z komputerem PC odbywa się w paśmie radiowym za pośrednictwem wieży nadawczo - odbiorczej podłączanej do komputera przez port USB. Komunikacja odbywa się na częstotliwości 38kHz, która zbliżona jest do częstotliwości używanej np. w pilotach telewizyjnych. Prędkość transmisji to 2400 bitów na sekundę, co oznacza że przesłanie jednego bitu trwa 417us. Logicznemu zeru odpowiada sygnał o długości 417us, logiczna jedynka to 417us „ciszy”. Ramka transmisyjna wygląda następująco: 1 bit startu – 8 bitów danych – 1 bit parzystości – 1 bit stopu.

Istotne jest, że zarówno nagłówek jak i treść przesyłanej informacji, nadawana jest w ramach zawierających taką samą ilość zer i jedynek, co pozwala odbiornikowi na korygowanie błędów transmisji spowodowanych promieniowaniem pochodzącym z otoczenia.

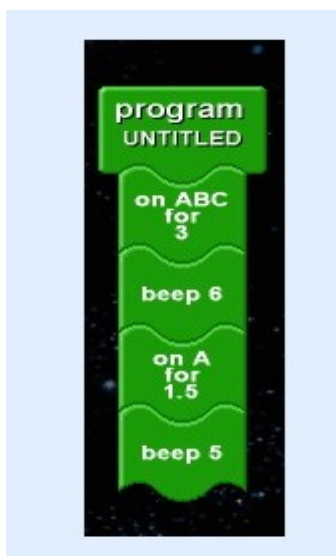


Rys. 5: Wieża nadawczo - odbiorcza wchodząca w skład zestawu [6].

Programowanie zestawu możliwe jest na kilka sposobów:

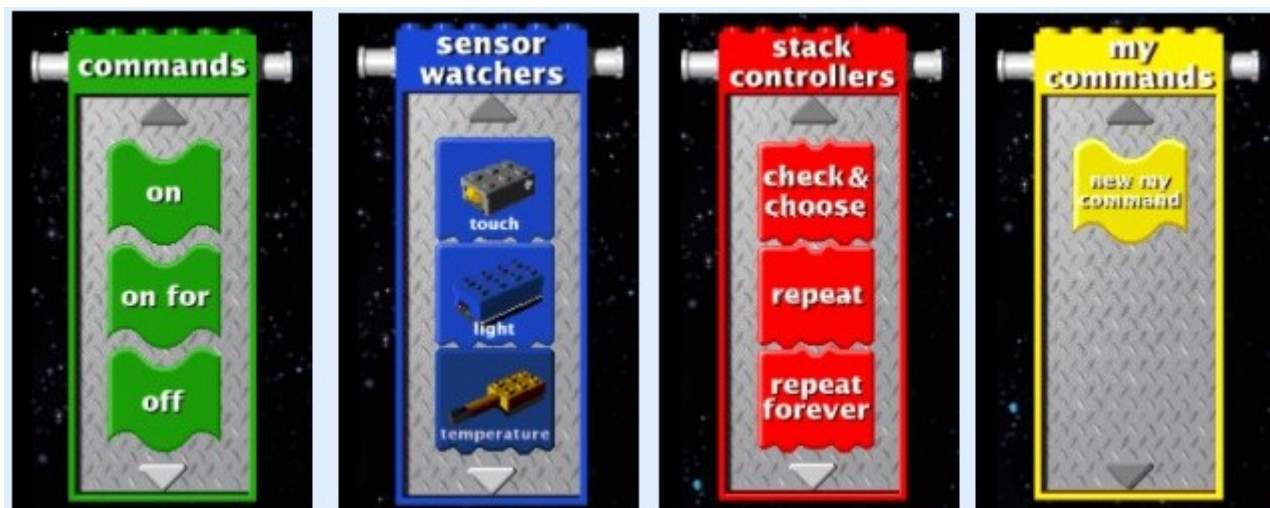
- programowanie za pomocą dostarczonego przez producenta oprogramowania
- programowanie za pomocą języków programowania stworzonych przez fanów zestawu

Programowanie za pomocą programu dostarczonego przez producenta (RCX CODE) jest bardzo intuicyjne. Nie umożliwia ono jednak pełnego wykorzystania zestawu. Polega ono na budowaniu algorytmów sterowania z klocków funkcyjnych, których właściwości może zmieniać programista na etapie programowania, a które umieszcza się jeden pod drugim zaczynając od klocka PROGRAM.



Rys. 6: Przykładowy fragment kodu w języku RCX CODE [3].

Poniższa ilustracja prezentuje większość dostępnych w programie klocków funkcyjnych:

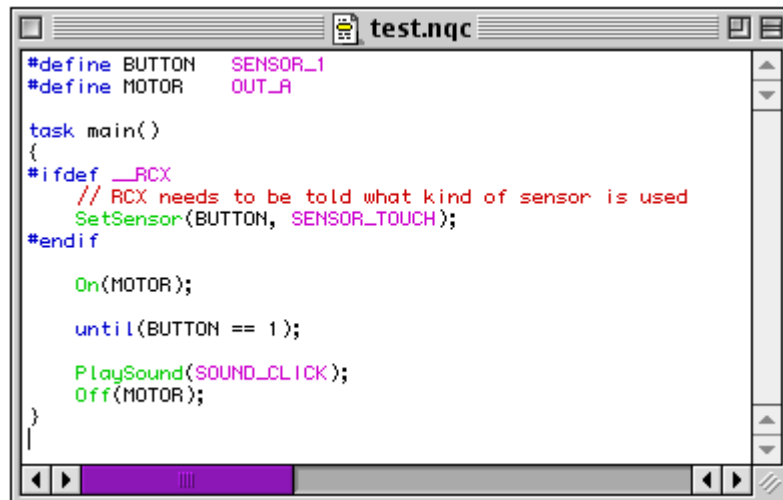


Rys. 7: Klocki funkcyjne dostępne w RCX CODE [3].

Jak widać, klocki te umożliwiają załączanie/wyłączanie silników, używanie sensorów, budowanie pętli, oraz tworzenie własnych komend. Jak jednak wspomniano już wcześniej, za pomocą tego narzędzia nie da się w pełni wykorzystać możliwości zestawu.

Wśród narzędzi programistycznych udostępnionych przez osoby nie związane zawodowo z LEGO na uwagę zasługuje program Nqcedit. Umożliwia on programowanie zestawu w języku NQC (Not Quite C), który, jak sama nazwa wskazuje, jest bardzo podobny do języka C.

Program ten daje programiście dużo większą swobodę w organizowaniu struktury algorytmów oraz wykorzystywania zasobów. Dodatkowo, narzędzi to w pełni zastępuje program dostarczony przez producenta obejmując oprócz kompilacji kodu na język zrozumiały dla modułu RCX, również obsługę wieży nadawczo – odbiorczej i przesyłanie kodu do jednostki sterującej.



```
#define BUTTON    SENSOR_1
#define MOTOR     OUT_A

task main()
{
#ifdef __RCX
    // RCX needs to be told what kind of sensor is used
    SetSensor(BUTTON, SENSOR_TOUCH);
#endif

    On(MOTOR);

    until(BUTTON == 1);

    PlaySound(SOUND_CLICK);
    Off(MOTOR);
}
```

Rys. 8: Okno programu Nqcedit wraz z fragmentem kodu w języku NQC [5].

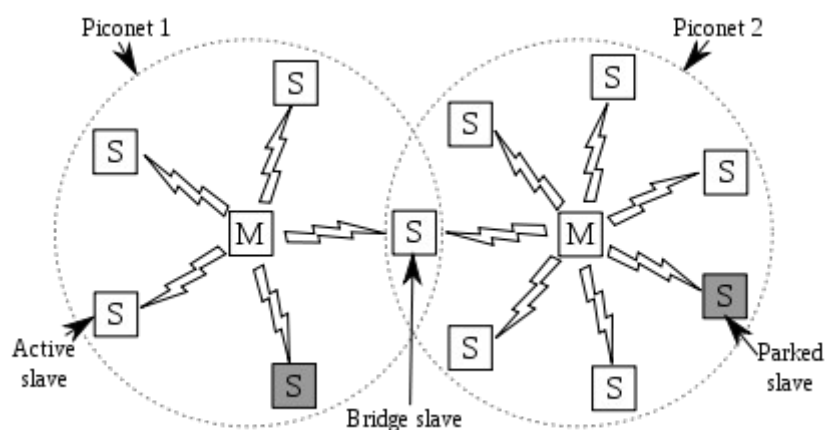
3. Komunikacja bezprzewodowa oparta na sieci Bluetooth.

„Bluetooth – technologia bezprzewodowej komunikacji krótkiego zasięgu pomiędzy różnymi urządzeniami elektronicznymi (...). Jest to darmowy standard opisany w specyfikacji IEE 802.15.1.” [4].

Nazwa technologii ma podłoże historyczne. Podobnie jak król duński – Harald Sinozęby (Blåtand) – zjednoczył plemiona duńskie i norweskie, tak technologia Bluetooth została stworzona do zintegrowania różnorodnych technologii w celu ułatwienia komunikacji pomiędzy nimi (komputery, telefony cyfrowe, aparaty fotograficzne i wiele innych). Za twórców technologii Bluetooth uważa się grupę SIG (Special Interest Group) założoną w 1994 roku przez firmy: L.M.Ericsson, IBM, Intel, Nokia oraz Toshiba. Celem powołania grupy było opracowanie bezprzewodowego łącza komunikacji pomiędzy telefonami komórkowymi. Technologia ta miała cechować się niewielkimi rozmiarami, niewielkim zasięgiem, nieznacznym poborem prądu, niską wartością mocy promieniowanej i niską ceną, co pozwoliłoby na stosowanie jej na szeroką skalę w aparatach telefonii cyfrowej.

Specyfikacja techniczna pierwszej wersji standardu Bluetooth (Bluetooth V1.0) pojawiła się w 1999 roku i stanowiła podstawę do dalszych nad systemem dla grupy standaryzacyjnej IEEE. Specyfikacja ta, obejmowała wszystkie warstwy systemu, począwszy od warstwy fizycznej aż po warstwę aplikacji.

Podstawową jednostką strukturalną sieci opartej na technologii Bluetooth jest tzw. pikosieć. Jest ona złożona z węzłów typu master oraz slave. Każda pikosieć może składać się z maksymalnie 1 węzła typu master oraz 7 węzłów typu slave. Każda pikosieć może zawierać dodatkowo do 255 węzłów pozostających w trybie wyczekiwania. Nie uczestniczą one w wymianie informacji, lecz pozostają zsynchronizowane z węzłem typu master, który może wysyłać do nich sygnały aktywujące i nawigacyjne. W jednym pomieszczeniu może występować kilka pikosieci. Ich połączenie również jest możliwe. Możemy tego dokonać za pomocą węzła typu bridge, który pracuje jako slave. Sieć złożona z dwóch lub więcej pikosieci nosi nazwę scatternet. Wymiana informacji zachodzi wyłącznie pomiędzy węzłem master a węzłem slave. Nie jest możliwa komunikacja slave to slave.



Rys. 9: Struktura sieci Bluetooth oparta na pikosieciach [4].

Najpopularniejszym formatem ramki transmisji w sieci Bluetooth jest ramka o następującej organizacji:

- kod dostępu
- nagłówek
- dane

Część ramki zawierająca kod dostępu złożona jest z 72 bitów, zawierających informacje o węźle typu master dla którego przeznaczona jest nadawana informacja w celu jednoznacznej identyfikacji na wypadek gdyby w pobliżu znajdowało się więcej niż jeden węzeł tego typu.

Następne pole w strukturze ramki zawiera nagłówek. Jest on złożony z 54 bitów, przy czym informacja przesyłana w tym miejscu powtarzana jest trzykrotnie (3 x 18 bitów) w celu korekcji ewentualnych błędów transmisji (jeżeli po stronie odbiorczej wszystkie 3 bity odpowiadające sobie są identyczne urządzenie przyjmuje odebraną wartość za prawdziwą, jeżeli któryś z bitów różni się od pozostałych, urządzenie przyjmuje wartość bitu jako wartość która odebrana została dwukrotnie).

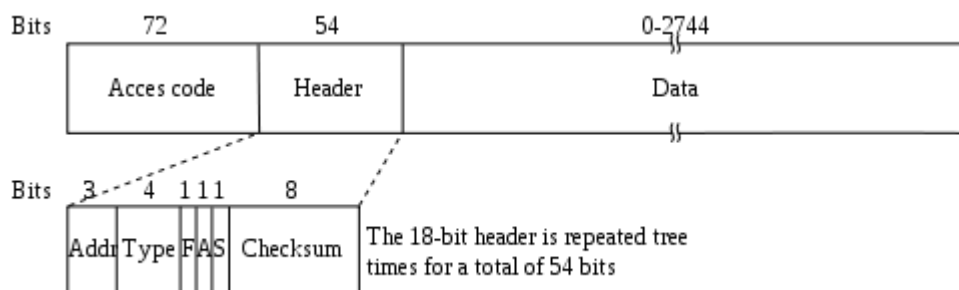
Nagłówek składa się z następujących stref:

- adres
- typ
- bit Flow
- bit Acknowledgment
- bit Sequence
- suma kontrolna

Strefy te zawierają odpowiednio informacje o:

- adresie jednego z 8 aktywnych węzłów sieci dla którego przeznaczona jest ramka
- typ ramki, ilość slotów w ramce oraz metodę korekcji błędów pola danych
- wypełnieniu bufora danych węzła slave i braku możliwości odebrania kolejnych informacji
- potwierdzeniu transmisji
- numerze ramki

Pole zawierające dane może zawierać 2777 bitów dla transmisji pięcioslotowej lub 240 bitów danych dla transmisji jednoslotowej.



Rys. 10: Organizacja ramki w sieci Bluetooth [4].

4. Projekt urządzenia do komunikacji bezprzewodowej opartej na interfejsie Bluetooth.

Głównym celem tego projektu jest zaprojektowanie modułu do komunikacji bezprzewodowej pomiędzy komputerem PC a komputerem zestawu LEGO. Obecna transmisja danych pomiędzy wyżej wymienionymi urządzeniami, jak zostało wspomniane już przedzej, odbywa się za pomocą komunikacji radiowej opartej na wieży nadawczo – odbiorczej promieniującej fale IR.

Rozwiązanie to posiada jedną, zasadniczą wadę, która wprowadza szereg ograniczeń na wykorzystanie zestawu. Otóż przy takim rozwiązaniu, wymiana informacji może się odbywać tylko wówczas, jeżeli kostka RCX i wieża podłączona do komputera PC znajdują się naprzeciwko sobie i nic nie zagraża przestrzeni pomiędzy nimi.

W związku z tym, nie jest możliwe przesyłanie informacji z komputera PC do komputera LEGO podczas ruchu robota. Wyjątkiem może być sytuacja gdy poruszamy „ręcznie” wieżę, lub gdy kable, za pośrednictwem których do kostki RCX podłączone są czujniki i silniki, są na tyle długie by komputer RCX nie był składową częścią robota, a jedynie komunikował się z nim przez wyżej wymienione kable. Oba przypadki wydają się zupełnie nieprzydatne a dodatkowo w przypadku z długimi kablami tracimy „bezprzewodowość” systemu.

Zastosowanie technologii Bluetooth pozwala na wyeliminowanie konieczności kierunkowego ustawiania komputera RCX tak aby widział wieżę nadawczą – odbiorczą podłączoną do komputera podczas ruchu robota. Daje nam to pełną swobodę w przesyłaniu informacji, rozkazów czy sterowania do robota będącego w ruchu.

4.1 Idea działania urządzenia.

Jak już wspomniano komunikacja z wykorzystaniem projektowanego modułu opierać się będzie na technologii Bluetooth. Aby było to możliwe moduł należy wyposażyć w następujące urządzenia:

- moduł Bluetooth,
- mikroprocesor,

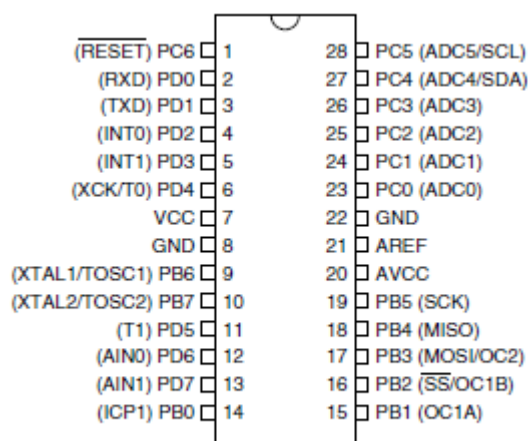
Zadaniem modułu Bluetooth będzie odbieranie informacji wysyłanych przez nadajnik Bluetooth zainstalowany w komputerze PC. Do projektu tego urządzenie wybrano moduł BTM-222. Jest to bardzo popularne i szeroko stosowane rozwiązanie w łączności bezprzewodowej Bluetooth. Następnie, pobrane rozkazy, przekazywać będzie za pośrednictwem interfejsu USART do mikrokontrolera zainstalowanego na płycie urządzenia. Mikrokontroler ten (ATmega8L), będzie pełnił rolę interpretatora poleceń odebranych z sieci oraz sterownika komputera RCX. Ponieważ wybrany do projektu urządzenia mikrokontroler nie posiada wyjść analogowych, konieczne jest zaprojektowanie interfejsu komunikacyjnego pomiędzy mikrokontrolerem a komputerem RCX. W przypadku tego projektu wybrano opcję polegającą na prostowaniu sygnału PWM, wystawianego przez ATmege, za pomocą filtra RC. Rozwiązanie to, mimo posiadania kilku wad, wydaje się najrozsądniejszym do tego zastosowania.

4.2 Wykorzystywane elementy.

Jak już wspomniano, kluczowymi elementami tego projektu są : moduł Bluetooth, mikroprocesor oraz filtr RC pełniący rolę wyjścia analogowego. W niniejszym podrozdziale zostaną one omówione w celu przybliżenia ich zasady działania oraz potwierdzenia słuszności wybrania konkretnych typów elementów.

4.2.1 Mikroprocesor Atmega8L.

Spośród wielu istniejących na rynku modeli oraz typów mikrokontrolerów, właśnie ten postanowiono użyć w projekcie modułu bezprzewodowego. Jedną z głównych, decydujących o wyborze tego modelu jest fakt, że pasuje on idealnie pod względem napięcia zasilania do wybranego modułu Bluetooth. Oba urządzenia zasilane są napięciem 3.3V co znacznie upraszcza projekt całego modułu ze względu na brak konieczności uwzględniania odrębnego poziomu napięcia. Dodatkowo, mikroprocesor Atmega8L zawiera praktycznie wszystko co potrzebne jest do tego projektu, patrząc na wymagania względem mikrokontrolera. Wyjątkiem jest brak wyjść analogowych. Podobnie sprawa wygląda jednak wśród mikroprocesorów oferowanych przez innych producentów. Oczywiście istnieją modele z wbudowanymi wyjściami analogowymi. Są one jednak dużo bardziej skomplikowane pod względem konstrukcyjnym a ich użycie, jak dla tego zastosowania, byłoby zbytecznym rozbudowywaniem całego modułu.



Rys. 11: Mikroprocesor ATmega8 wraz z opisem wyprowadzeń [7].

Powyżej zaprezentowano schemat wykorzystywanego w projekcie mikroprocesora, oraz funkcje pełnione przez konkretne wyprowadzenia. Z punktu widzenia projektu, przydatne będą tylko niektóre z nich. Zanim jednak do nich dojdziemy, opiszmy po krótko każde z wyprowadzeń:

- RESET / PC6 – w zależności od zaprogramowania Fuse – bitów mikrokontrolera, wyprowadzenie to może pełnić funkcję resetu (domyślnie) lub funkcję portu wejściowo – wyjściowego, reset mikrokontrolera aktywujemy impulsem o wartości logicznej 0,
- RXD / PD0 – w zależności od konfiguracji rejestrów odpowiedzialnych za obsługę modułu USART mikrokontrolera, wyprowadzenie to może pełnić rolę odbiornika interfejsu USART (receiver) lub portu wejściowo – wyjściowego,

- TXD / PD1 – podobnie jak poprzednie wyprowadzenie, w zależności od konfiguracji rejestrów odpowiedzialnych za moduł USART może on pełnić rolę nadajnika interfejsu USART (transmitter) lub portu wejściowo – wyjściowego,
- INT0/PD2 oraz INT1/PD3 – oba te porty w zależności od konfiguracji mikrokontrolera mogą pełnić rolę wejść dla zewnętrznych przerwań lub portów wejściowo – wyjściowych,
- (XCK/T0) / PD4 – wyprowadzenie to, oprócz funkcji standardowego portu wejściowo – wyjściowego, może być portem komunikacyjnym dla zewnętrznego zegara dla interfejsu USART, lub źródła impulsów dla licznika COUNTER0,
- VCC – wyprowadzenie do którego podłączamy napięcie zasilania,
- GND – uziemienie układu,
- (XTAL1 / TOSC1) / PB6 oraz (XTAL2 / TOSC2) / PB7 – wyprowadzenia te pełnią rolę wejść dla zewnętrznych oscylatorów kwarcowych taktujących pracę mikrokontrolera (jako piny XTAL1, XTAL2), lub do taktowania TIMERA2 (jako piny TOSC1, TOSC2), pin ten może też pełnić funkcję standardowego portu wejściowo – wyjściowego,
- T1 / PD5 – wyprowadzenie służące do taktowania TIMERA1 lub pełniące funkcję portu wejściowo – wyjściowego,
- AIN0 / PD6 - wejście nieodwracające komparatora analogowego. Należy pamiętać, aby wyłączyć wewnętrzne podciąganie pull-up gdy pin pełni tę funkcję. Wyprowadzenie może również pełnić funkcję portu wejściowo – wyjściowego,
- AIN1 / PD7 - wejście odwracające komparatora analogowego. Należy pamiętać, aby wyłączyć wewnętrzne podciąganie pull-up gdy pin pełni tę funkcję. Wyprowadzenie może również pełnić funkcję portu wejściowo – wyjściowego,
- ICP1 / PB0 – może pełnić funkcję wejścia przechwytywania dla TIMER1 lub standardowego portu wejściowo – wyjściowego,
- (ADC5 / SCL) / PC5 – oprócz funkcji standardowego portu wejściowo – wyjściowego, wyprowadzenie to może pełnić funkcję wejścia kanału piątego konwertera analogowo – cyfrowego, lub linii zegarową interfejsu TWI. Aby przypisać do pinu funkcję linii zegarowej interfejsu TWI należy ustawić bit TWEN w rejestrze TWCR,
- (ADC4 / SDA) / PC4 – podobnie jak poprzednie wyprowadzenie może ono oprócz funkcji portu wejściowo – wyjściowego pełnić funkcję wykorzystywaną w interfejsie TWI – linię danych (wejście / wyjście). Aby przypisać pinowi tą funkcję należy ustawić bit TWEN w rejestrze TWCR. Jest to również wejście kanału czwartego, konwertera analogowo – cyfrowego,
- ADC3 / PC3, ADC2 / PC2, ADC1 / PC1 oraz ADC0 / PC0 – oprócz funkcji portów wejściowo – wyjściowych, pełnią funkcję wejść kanałów 1,2,3 i 4 konwertera analogowo – cyfrowego,
- AREF – wyprowadzenie do którego podłączamy napięcie referencyjne przetwornika analogowo – cyfrowego.
- AVCC – napięcie zasilania dla przetwornika analogowo – cyfrowego,
- SCK / PB5 – wyprowadzenie pełni rolę wyjścia zegara master, lub wejście zegara slave dla magistrali SPI, pin może służyć również jako port wejściowo – wyjściowy,
- MISO / PB4 – Master Input Slave Output – wyprowadzenie pełni rolę wejścia danych dla konfiguracji master lub wyjścia dla konfiguracji Slave magistrali SPI, może pracować również jako port wejściowo – wyjściowy,
- (MOSI / OC2) / PB3 – Master Output Slave Input – wyprowadzenie pełni rolę wyjścia danych dla konfiguracji master lub wejścia dla konfiguracji Slave magistrali SPI. Innym zastosowanie wyprowadzenia jest wyjście układu porównania COUNTER2, lub wyjście sygnału PWM. Może pracować również jako port wejściowo – wyjściowy,

- (SS / OC1B) / PB2 – oprócz roli portu wejściowo – wyjściowego, wyprowadzenie może pełnić rolę wejścia wyboru slave (Slave Select), wyjścia układu porównania licznika COUNTER1, lub wyjścia PWM,
- OC1A / PB1 – wyprowadzenie może pełnić funkcję wyjścia układu porównania COUNTER1 lub wyjścia PWM. Może być również portem wejściowo – wyjściowym.

Jak więc widzimy, wyposażenie tego mikrokontrolera pozwala na zastosowanie go w wielu projektach.

Z punktu widzenia tej pracy, potrzebne nam będą następujące funkcje mikroprocesora:

- interfejs USART
- tryb PWM licznika

Dodatkowo urządzenie będzie wyposażone w możliwość podłączenia programatora SPI w celu zmiany programu mikrokontrolera, co poszerza jego funkcjonalność (można m.in. zmienić ilość rozpoznawalnych i rozróżnianych przez mikrokontroler rozkazów przesłanych z komputera PC).

Aby wykorzystać wszystkie wyżej wymienione narzędzia skorzystano z następujących wyprowadzeń:

- OC1A / PB1 - jako wyjście sygnału PWM sterujące komputerem RCX LEGO,
- TXD / PD1 - jako wyjście interfejsu USART,
- RXD / PD0 - jako wejście interfejsu USART,
- SCK / PB5, MISO / PB4, MOSI / PB3 – do komunikacji z programatorem SPI.

Dodatkowo wykorzystany zostanie pin ADC0 / PC0 w konfiguracji jako port wyjściowy, do którego podłączona zostanie dioda sygnalizująca poprawne działanie całego modułu.

4.2.2 Moduł Bluetooth BTM-222

Kolejnym urządzeniem wykorzystanym do projektu modułu komunikacji bezprzewodowej pomiędzy komputerem PC a komputerem RCX LEGO jest moduł Bluetooth BTM-222.

Moduł ten został wybrany ze względu na swoją popularność w podobnych zastosowaniach. Duże zainteresowanie modulem, oraz częste jego wykorzystywanie skutkuje sporą ilością materiałów o samym module jak i projektach z jego użyciem, co pomogło w tworzeniu tego urządzenia.

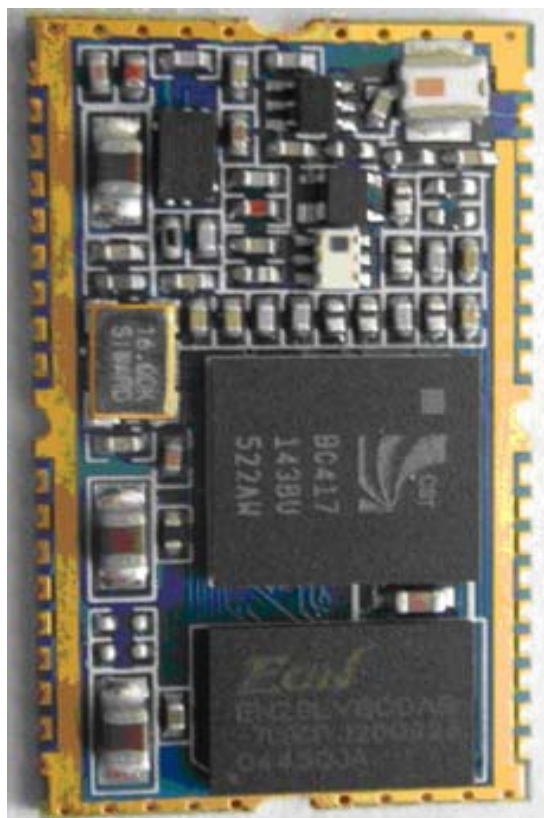
Jest to moduł Bluetooth oznaczony jako moduł klasy I co oznacza iż może on pracować na zasięgu do 100m, a jego moc nadawcza może wynosić do +18dBm. Napięcie zasilania modułu wynosi 3,3V co idealnie pasuje do poziomu zasilania użytego mikrokontrolera. Według dokumentacji technicznej, moduł ten posiada możliwość komunikacji poprzez interfejs UART oraz USB. Podczas zdobywania informacji na temat tego modułu nie udało się odnaleźć żadnych projektów wykorzystujących to urządzenie komunikujących się poprzez USB. Co więcej, większość osób wykorzystujących go, twierdziło o niemożliwości korzystania z wyżej wymienionego interfejsu komunikacyjnego. Niemniej jednak nie jest to problemem, gdyż w tym projekcie wykorzystane zostaną piny układu odpowiedzialne za komunikację po interfejsie UART.

Ustawienia modułu możemy zmieniać za pośrednictwem komend AT przesyłanych do urządzenia. W przypadku tej pracy, nie jest konieczne modyfikowanie ustawień fabrycznych (prędkość transmisji – 19200 bps, 8 bitów danych, brak bitu parzystości, 1 bit stopu) gdyż podobne ustawienia można dokonać na mikrokontrolerze używanym w tej pracy.

Można jednak wymienić niektóre z komend AT sterujących tym modulem:

- ATI – komenda nakazująca podanie wersji Firmware zainstalowanego w module,
- ATK(0,1,?) - komenda ustawiająca 1, 2 bity stopu lub podająca obecne ustawienie(?),
- ATL(1,2...7,?) - komenda ustawiająca prędkość transmisji lub podająca obecne ustawienie,
- ATM(0,1,2,?) - komenda ustawiająca sprawdzanie bitu parzystości/nieparzystości, wyłączająca kontrolę lub sprawdzająca obecne ustawienia,
- ATZ0 – przywrócenie ustawień fabrycznych modułu.

Aby wysłać komendę AT do modułu należy najpierw napisać program, który będzie wysyłać kolejne znaki komendy poroździelane krótkimi przerwami czasowymi (są one różna w zależności od tego czy wysyłamy polecenia w trybie z echem i bez echa i powinny wynosić odpowiednio min. 4-5ms i min.25 ms). Bardzo ważne jest również, aby każda komenda zakończona była znakiem ENTER, gdyż jest to konieczne dla powodzenia całej operacji.



Rys. 12: Moduł Bluetooth BTM-222 [8].

Moduł ten posiada wiele wyprowadzeń, lecz większość z nich nie jest aktywna co skutkuje brakiem możliwości ich wykorzystania. W projekcie tym do modułu (oprócz pinów zasilania i masy) wykorzystane zostaną piny odpowiedzialne za komunikację UART oraz pin pełniący funkcję wejścia antenowego.

Mówiąc o antenie należy wspomnieć, że do tego zastosowania nie jest ona właściwie konieczna. Na rynku dostępne są różnego rodzaju anteny Bluetooth lecz, w tym konkretnym zastosowaniu, można obyć się bez niej, podłączając pin do krótkiego kawałka drutu lub ścieżki na płytce drukowanej urządzenia. W przypadku tego projektu zastosowano krótki odcinek drutu miedzianego.

4.3 Interfejs komunikacyjny pomiędzy mikroprocesorem a modulem RCX 2.0.

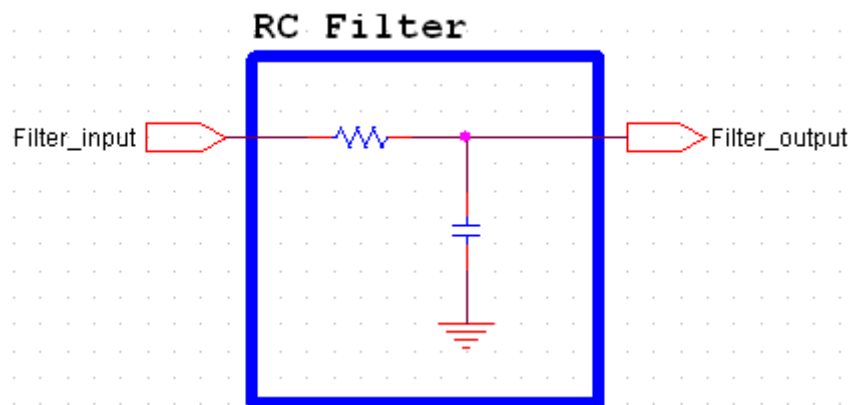
Jak już wcześniej wspomniano, mikrokontroler wykorzystany w tym projekcie nie posiada wbudowanego przetwornika cyfrowo – analogowego pełniącego funkcję wyjścia. Konieczne jest więc zatem zaprojektowanie odrębnego elementu pełniącego tą rolę. Ze względu na prostotę realizacji zdecydowano o realizacji wyjścia analogowego poprzez prostowanie sygnału PWM filtrem biernym RC.

Rozwiązanie to jest szeroko stosowane ze względu na prostotę oraz niskie koszty realizacji.

Jednak sam filtr nie rozwiązuje jeszcze kwestii połączenia mikrokontrolera z komputerem RCX.

Wykonane pomiary wykazały, że przy podawaniu na wejście kostki RCX napięcia 1,2 V skutkuje wyświetleniem się wartości odczytu 100%. Oznacza to, że jest to najwyższe napięcie jakie możemy zastosować do sterowania RCX.

Przy 100% wypełnieniu sygnału PWM, na wyjściu filtru otrzymamy napięcie zasilania mikrokontrolera równe 3,3V. W celu zmniejszenia tego napięcia należy więc zastosować rezystor włączony szeregowo za filtrem RC.



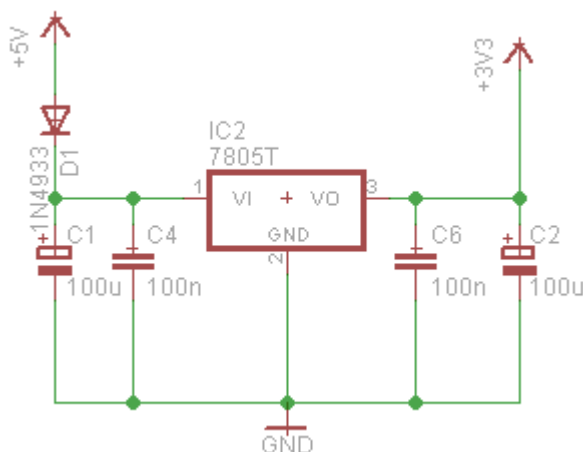
Rys. 13: Filtr RC służący do prostowania sygnału PWM [10].

Zgodnie z informacjami zawartymi w [8], aby prostowanie sygnału PWM poprzez filtr RC było skuteczne należy tak dobrać częstotliwość PWM oraz elementy filtru, aby stała czasowa filtru była około 100 razy wyższa niż okres przebiegu PWM.

4.4 Schemat połączeń elektrycznych.

Poniżej zaprezentowane zostaną schematy połączeń pomiędzy poszczególnymi elementami, jak również schemat zasilania układu. Schematy zostały wykonane w wersji testowej programu CADSOFT EAGLE wersji 5.10.0 dla systemu Windows (symbole elementów na schematach mogą różnić od użytych w projekcie ze względu na brak biblioteki z używanymi elementami, są one jednak zgodne pod względem wymiarów z oryginalnymi, występującymi w pracy elementami).

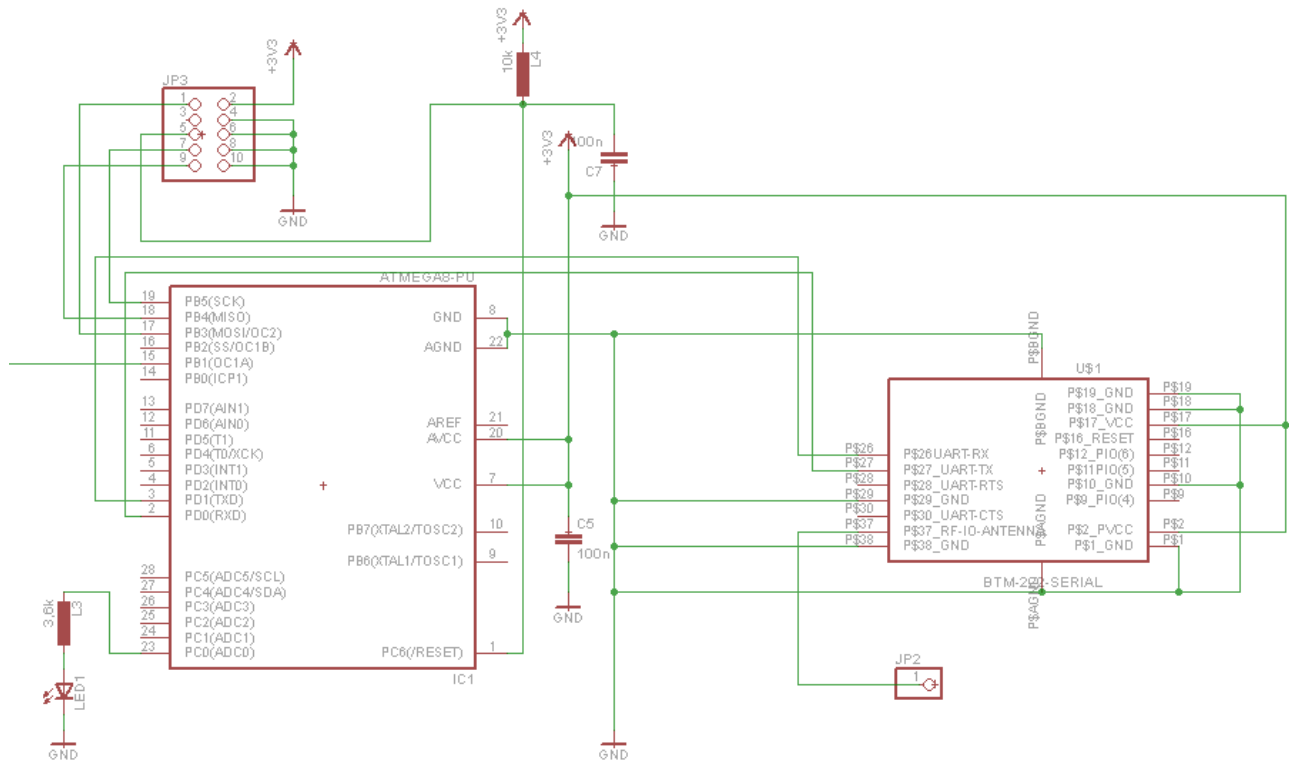
- schemat zasilania:



Rys. 14: Schemat zasilania układu.

W projekcie użyto stabilizatora o symbolu LM007T dającego na wyjściu napięcie na poziomie 3,3V. Kondensatory chronią stabilizator przed wzbudzaniem w razie zasilania innego niż bateryjne.

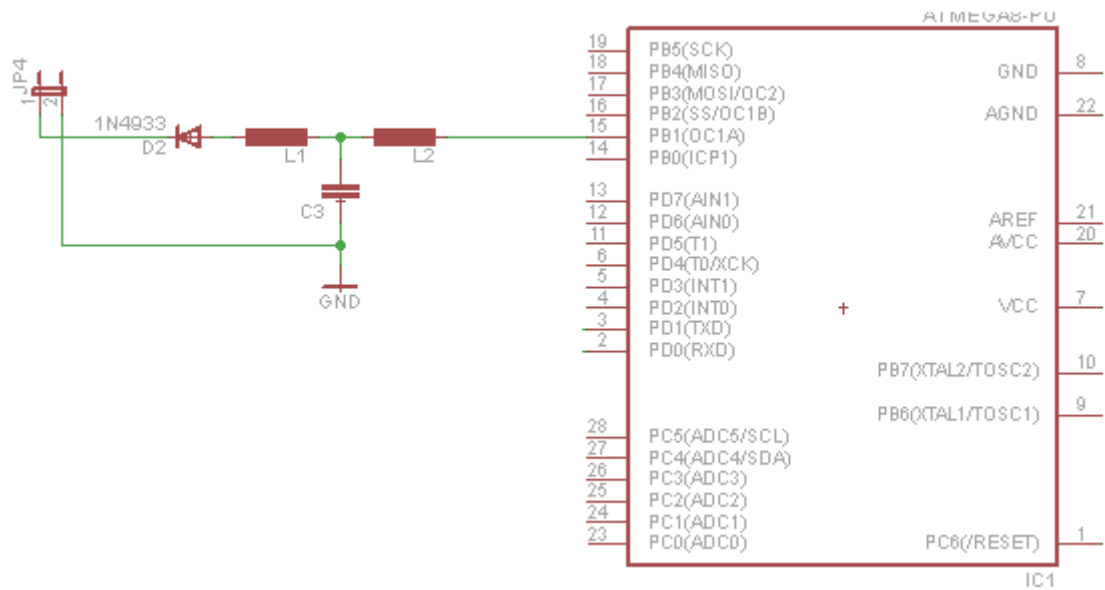
– mikrokontroler Atmega8L– BTM-222:



Rys. 15: Schemat połączenia mikrokontrolera z modulem Bluetooth.

Powyższy schemat obrazuje sposób podłączenia mikrokontrolera Atmega8L do modułu BTM-222. Jest na nim zawarta również dioda sygnalizująca poprawne działanie układu, gniazdo programatora, oraz miejsce na podłączenie anteny dla modułu Bluetooth.

- filtr RC wraz z rezystorem dopasowującym poziom napięć mikrokontrolera i komputera RCX:



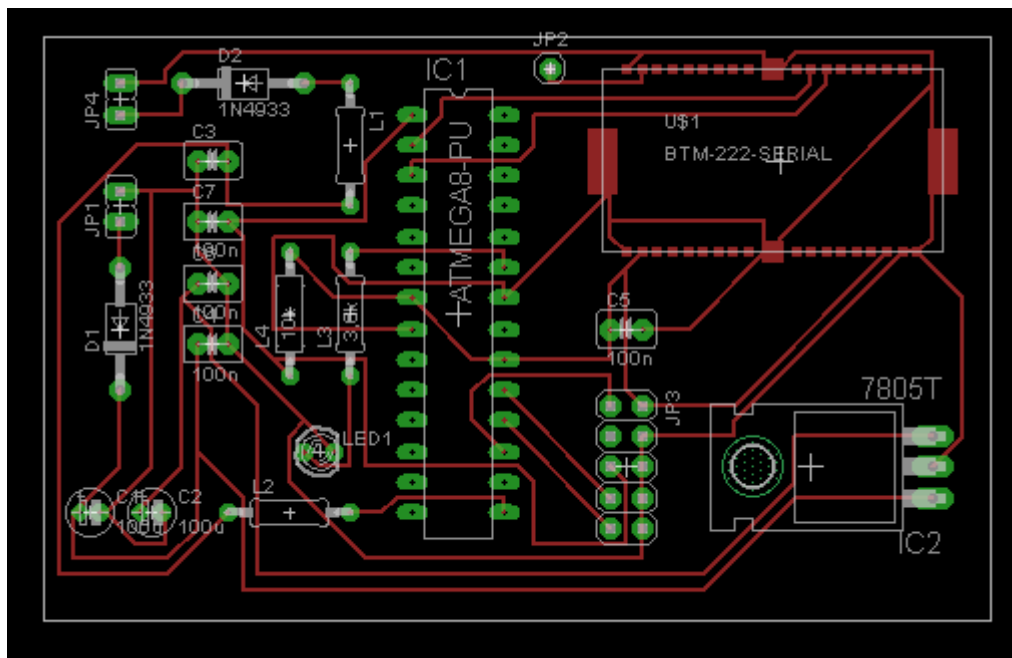
Rys. 16: Schemat połączenia mikrokontrolera z filtrem RC.

Jak widzimy, w schemacie uwzględniono diodę zatrzymującą napięcie wystawiane przez kostkę RCX (działanie wejść komputera RCX w trybie aktywnym zostało opisane we wcześniejszym rozdziale).

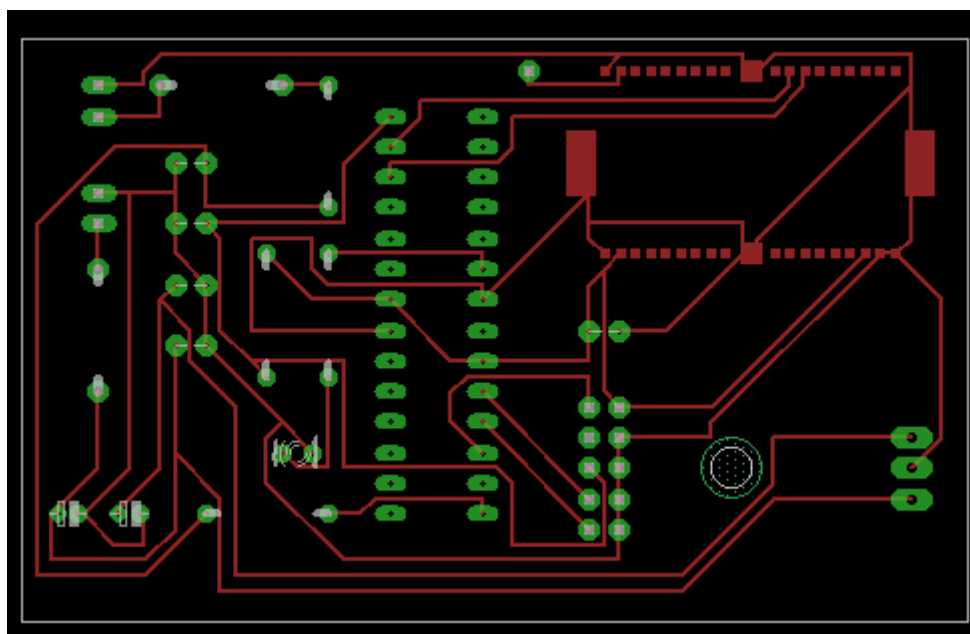
4.5 Projekt płytki drukowanej.

Podobnie jak schematy połączeń, również projekt płytki drukowanej został wykonany w programie CADSOFT EAGLE. W załączniku znajduje się projekt płytki w większym rozmiarze.

Ze względu na ograniczenia w wykonaniu płytki, jest ona jednostronna.



Rys. 17: Projekt płytki drukowanej.

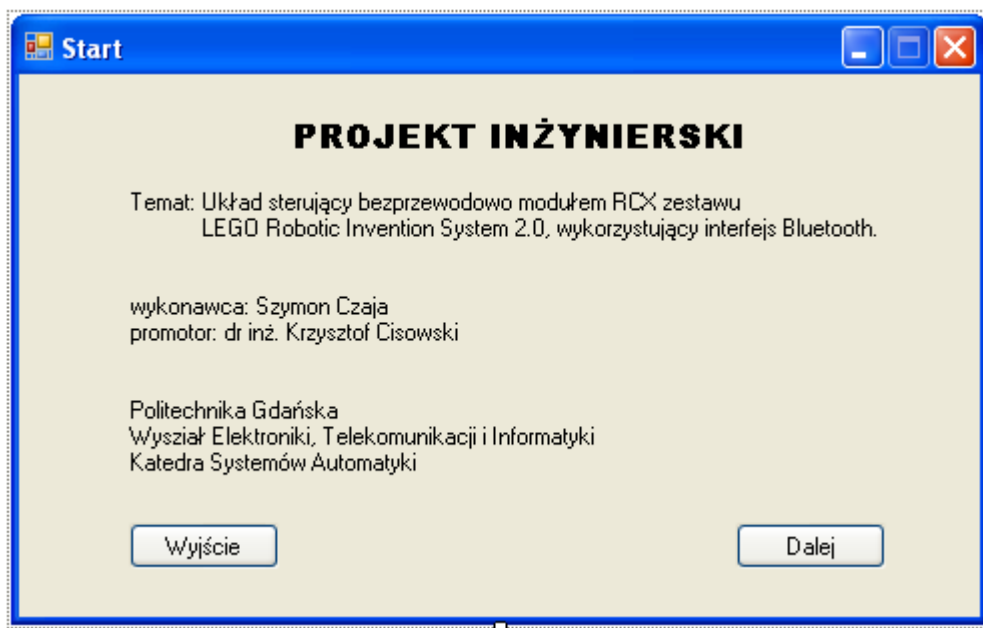


Rys. 18: Rozkład ścieżek i otworów pod elementy.

5. Oprogramowanie komputera PC

Do napisania oprogramowania umożliwiającego przesyłanie informacji z komputera PC to modułu BTM-222 użyto programu Microsoft Visual Studio 2008 na licencji studenckiej. Program napisano w języku C#. Wyżej wspomniane narzędzie programistyczne w dość łatwy sposób pozwala budować interfejsy graficzne do komunikacji użytkownika z komputerem, oraz obsługę wysyłania danych przez port szeregowy.

Po włączeniu programu użytkownikowi ukazuje się okno startowe zawierające informacje odnośnie projektu.



Rys. 19: Okno startowe programu na PC.

Zawiera on informacje takie jak: temat pracy, wykonawcę i opiekuna pracy. Naciśnięcie przycisku „Wyjście” powoduje zamknięcie aplikacji. Natomiast naciśnięcie przycisku „Dalej” powoduje zamknięcie obecnego okna i wyświetlenie kolejnego.

Kolejne okno programu umożliwia użytkownikowi:

- nadanie nazw poleceniom,
- wybranie portu COM skojarzonego z modulem,
- nawiązanie próbnego połączenia,
- przesyłanie rozkazów przypisanych do poszczególnych przycisków.
- Zakończenie działania aplikacji.

Konfig

Numer portu COM skojarzonego z modulem Bluetooth:

polecenie 1:

polecenie 2:

polecenie 3:

polecenie 4:

polecenie 5:

polecenie 6:

polecenie 7:

polecenie 8:

polecenie 9:

polecenie 10:

Rys. 20: Okno konfiguracyjne programu.

Poniżej zaprezentowano fragment kodu odpowiedzialnego za nadanie i zmianę nazwy polecenia 1.

```
private void button2_Click(object sender, EventArgs e)
{
    label13.Text = textBox2.Text;
    button23.Text = label13.Text + " (" + label133.Text + ")";
    textBox2.Visible = false;
    label13.Visible = true;
}
private void label13_Click(object sender, EventArgs e)
{
    label13.Visible = false;
    textBox2.Visible = true;
}
```

Jak widzimy, nadanie nazwy powoduje zniknięcie pola tekstowego przeznaczonego do jej wpisania i wyświetlenie się w tym miejscu napisu wprowadzonego przez użytkownika. Nadanie nazwy dokonujemy poprzez kliknięcie przycisku “zmień”

Sytuacja wygląda identycznie w przypadku pozostałych 9 poleceń.

Kolejny fragment kodu prezentuje wybór portu COM i jego zmianę:

```
private void button1_Click(object sender, EventArgs e)
{
    label2.Text = textBox1.Text;
    textBox1.Visible = false;
    label2.Visible = true;
}

private void label2_Click(object sender, EventArgs e)
{
    label2.Visible = false;
    textBox1.Visible = true;
}
```

Sytuacja wygląda podobnie jak przy nadawaniu nazw poleceniom.

W celu nawiązania połączenia z mikrokontrolerem użyto komponentu serialPort1. Jest on skonfigurowany tak, aby pasował do fabrycznych ustawień modułu Bluetooth.

Po nadaniu nazw poleceniom i wybraniu portu możliwe jest nawiązanie połączenia poprzez wciśnięcie przycisku “Połącz z modulem”.

```
//*****Obsługa połączenia z uC*****  
  
private void button22_Click(object sender, EventArgs e)  
{  
    serialPort1.PortName = label2.Text;  
    serialPort1.Write ();  
  
    //wysłanie do portu informacji o skończeniu konfiguracji  
  
}
```

Powyższy fragment kodu obrazuje realizację tej operacji.

Label2.text zawiera nazwę portu COM wraz z numerem wprowadzonym przez użytkownika. Polecenie serialPort1.Write (); odpowiada za wysłanie do portu informacji umieszczonej pomiędzy nawiasami ().

Identycznie sytuacja wygląda podczas wysyłania poleceń sterujących komputerem RCX.

6. Podsumowanie

Podsumowując powyższy projekt trzeba wspomnieć o trudnościach napotkanych podczas jego realizacji. Wśród nich warto wspomnieć takie jak :

- niewiele źródeł informacji o działaniu wejść komputera RCX,
- technologię wykonania modułu BTM-222 (smd).

Zwłaszcza druga z wymienionych tutaj przeszkód zadecydowała o niezrealizowaniu fizycznej wersji modułu. Brak fizycznej realizacji zaowocował niemożliwością przetestowania programu na komputer PW co skutkuje dołączeniem do pracy jedynie fragmentów kodu opisujących idee działania programu, a nie pełną jego postać.

Trzeba też jednak wspomnieć, że projekt ten łączy w sobie wiele zagadnień, których poznanie wymagało poszerzenia wiedzy na wielu horyzontach (projektowanie i wykonywanie schematów i płytek drukowanych, programowania w języku C# oraz C). Wszystko to poskutkowało znacznym poszerzeniem wiedzy w wyżej wymienionych dziedzinach.

7. Literatura

- [1] <http://konkurs.best.krakow.pl/2006/legomindstorms.php>
- [2] <http://www.crynwr.com/lego-robotics/>
- [3] <http://gajdaw.pl/roboty-lego-mindstorms/rcx-code/p1.html>
- [4] <http://pl.wikipedia.org/wiki/Bluetooth>
- [5] <http://homepage.mac.com/rbate/MacNQC/index.html>
- [6] <http://www.rapidonline.com/1/1/indexe220.html>
- [7] nota katalogowa AVR ATmega8
- [8] nota katalogowa BTM-222
- [9] <http://www.elektroda.pl/rtvforum/viewtopic.php?t=1433390&highlight=>
- [10] <http://dev.emcelettronica.com>
- [11] Rafał Baranowski, "Mikrokontrolery AVR ATmega w praktyce".