

# **SYSTEMY WBUDOWANE**

## **Dokumentacja projektu**

**Grupa C15**

Autorzy:

- \* Nowicki Artur 224388 (Lider)
- \* Guzek Paweł 224304
- \* Maksajda Michał 224369

30.08.2020

# Spis treści

1	Wykorzystane funkcjonalności .....	3
2	Zakres obowiązków i udział procentowy.....	3
3	Analiza FMEA.....	4
4	Instrukcja użytkownika.....	6
5	Sposób odczytu danych z czujników .....	7
5.1	Akcelerometr/Żyroskop LSM6DS3 .....	7
5.1.1	Opis interfejsu komunikacyjnego (SPI): .....	7
5.1.2	Opis odczytu danych z urządzenia pomiarowego: .....	8
5.1.3	Wbudowane funkcje (krokomierz) (roz. 6 [2]):.....	9
5.2	Magnetometr AK09918.....	10
5.2.1	Opis interfejsu komunikacyjnego I2C:.....	10
5.2.2	Tryby pracy czujnika: .....	12
5.2.3	Opis odczytu danych z urządzenia pomiarowego: .....	13
5.3	Czytnik kart NFC PN544 .....	14
5.3.1	Opis interfejsu komunikacyjnego UART:.....	14
5.3.2	Logiczna warstwa łącza (Logic Layer Link / LLC): .....	15
5.3.3	Zarządzenie PN544: .....	15

# 1 Wykorzystane funkcjonalności

Nazwa czujnika	Oznaczenie	Interfejs	Funkcja
Akcelerometr	LSM6DS3	SPI	Sterowanie kulą po ekranie.
Żyroskop	LSM6DS3	SPI	Położenie panoramy w tle.
Krokomierz	LSM6DS3	SPI	Modyfikacja ilości otrzymywanych punktów.
Magnetometr	AKM09918	I2C	Sterowanie kulą po ekranie / spełnienie warunku startowego gry.
Czytnik kart NFC	PN544	UART	Aktywacja trybu „oszusta”.

# 2 Zakres obowiązków i udział procentowy

Autor	Zadania	Udział procentowy
Artur Nowicki	<ul style="list-style-type: none"><li>• Ruchomy obraz panoramiczny w tle na podstawie odczytów z żyroskopu.</li><li>• Dokumentacja – rozdziały: 1, 2 i 5.1.</li><li>• Formatowanie dokumentacji.</li></ul>	34%
Paweł Guzek	<ul style="list-style-type: none"><li>• Aktywacja trybu „oszusta” przy odczycie odpowiedniego tagu NFC.</li><li>• Zmiana ilości otrzymywanych punktów przez gracza na podstawie funkcji wbudowanych akcelerometru.</li><li>• Dokumentacja – rozdziały: 4 i 5.3.</li></ul>	33%
Michał Maksajda	<ul style="list-style-type: none"><li>• Poruszanie się kulki wg odczytu z akcelerometru i magnetometru.</li><li>• Spełnienie warunku startowego gry na podstawie wskazania z kompasu.</li><li>• Dokumentacja – rozdziały: 3 i 5.2.</li></ul>	33%

### 3 Analiza FMEA

Awaria	Znaczenie	Skala uciążliwości	Reakcja	Prawdopodobieństwo wystąpienia
Interfejs I2C	Krytyczny	10	Brak możliwości komunikacji z magnetometrem.	0.01
Interfejs SPI	Krytyczny	10	Brak możliwości komunikacji z akcelerometrem, żyroskopem oraz krokomierzem.	0.01
Interfejs UART	Średni	6	Brak możliwości komunikacji z kontrolerem NFC.	0.01
Kontroler NFC	Średni	4	Brak możliwości aktywowania trybu oszusta, który udostępnia dodatkowe funkcjonalności podczas gry.	0.02
Akcelero metr	Wysoki	9	Brak danych z akcelerometru uniemożliwia sterowanie kulką po ekranie oraz powoduje brak aktualizacji danych krokomierza.	0.02
Magneto metr	Wysoki	9	Utrudniona interakcja gracz – urządzenie (np. brak możliwości spełnienia warunku startowego gry), brak reakcji w grze na aktualizujące się dane z magnetometru, a zatem niemożliwe poruszanie kulki.	0.02

<b>Awaria</b>	<b>Znaczenie</b>	<b>Skala uciążliwości</b>	<b>Reakcja</b>	<b>Prawdopodobieństwo wystąpienia</b>
Żyroskop	Niski	2	Ruchomy obraz panoramiczny w tle przestaje funkcjonować ze względu na brak dostarczanych danych z żyroskopu.	0.02
Awaria ekranu (digitizer) – brak reakcji na dotyk	Niski	3	Brak możliwości skorzystania z funkcjonalności dodatkowej – bonusu punktowego po kliknięciu w ekran.	0.01

## 4 Instrukcja użytkownika

Po uruchomieniu aplikacji użytkownik jest pytany o przyznanie uprawnienia do aktywności fizycznej, w przypadku odrzucenia prośby o uprawnienie użytkownik będzie dostawał obniżoną ilość punktów, aby rozpocząć rozgrywkę należy skierować urządzenie w kierunku północnym, wysyłane jest wtedy powiadomienie o pomyślnym rozpoczęciu gry. Użytkownik za pomocą rotacji urządzenia porusza kulka po ekranie. Jego zadaniem jest utrzymanie kulki w żółtej strefie. W momencie, gdy kulka znajduje się w strefie, gracz otrzymuje punkty.

Ilość kroków liczona jest od ostatniego restartu urządzenia. Na wartość ruchu ma wpływ poruszanie się z włączoną aplikacją. Liczba przyznawanych punktów liczona jest ze wzoru:

$$P = P_p + D * \max \left( 1, \left( \frac{K}{200} \right) - R \right),$$

P – nowa ilość punktów

$P_p$  – poprzedni stan punktów

D - domyślna ilość punktów = 1

K - ilość kroków

R - ruch

Punkty przyznawane są z każdą aktualizacją wartości orientacji, ponadto, jeżeli gracz znajduje się wewnątrz strefy generowana jest pseudolosowa wartość z zakresu  $\langle 0f; 1.0f \rangle$ , jeżeli jest ona mniejsza niż 0.01f dochodzi do przesunięcia strefy, w której można zdobyć punkty. Gracz za pomocą tagu (karty) NFC o określonym numerze seryjnym może włączyć tryb oszusta, w którym po dotknięciu ekranu przyznawana jest losowa wartość punktów z zakresu  $\langle 20; 100 \rangle$  dodatkowo wyłączany jest pęd kulki.

## 5 Sposób odczytu danych z czujników

### 5.1 Akcelerometr/Żyroskop LSM6DS3

Dokumentacja urządzenia pomiarowego:

[1] [LSM6DS3](#)

Nota aplikacyjna:

[2] [LSM6DS3: always-on 3D accelerometer and 3D gyroscope](#)

#### 5.1.1 Opis interfejsu komunikacyjnego (SPI)

Jednym ze sposobów komunikowania się z urządzeniem pomiarowym LSM6DS3 jest interfejs SPI. Dokonywane jest to przez ustawienie pinu #12 (CS) na 0. Do komunikacji wykorzystywane są piny: #1 (SDO - wyjście), #12 (CS - aktywacja), #13 (SPC - zegar) i #14 (SDI - wejście) (tabela 2. [1]). Określa to bit SIM ustawiony na 0 (domyślnie) w rejestrze CTRL3\_C (4-wire interface) (roz. 9.14 [1]).

Interfejs SPI działa w trybie “master-slave”. W tym przypadku “slave” odnosi się do urządzenia pomiarowego, a “master” do urządzenia pobierającego dane z urządzenia pomiarowego. Urządzenie określane jako “master” inicjuje transmisję poprzez:

- ustawienie parametrów komunikacji (roz. 4.4.1 [1]):
  - polaryzacja i faza zegara SPI “Mode 3”: CPOL = 1, CPHA = 1,
  - częstotliwość 10 MHz: dzielnik zegara SPI (PRESCALER),
- rozpoczęcie transmisji: ustawienie pinu #12 (CS) na 0.

Po wymianie wszystkich bitów danych “master” zmienia stan linii CS na wysoki.

Interfejs pracuje w trybie “Mode 3”. Oznacza to, że polaryzacja i faza zegara są ustawione tak, że piny na których przesyłane są dane do i z urządzenia (SDI i SDO) są ustawiane na opadającym zboczu zegara (SPC), a odczytywane na rosnącym.

Transmisja może odbyć się minimum 20 ns po ustawieniu pinu CS na 0 (roz. 4.4.1 [1]). Przez interfejs SPI przesyłane jest minimum 16 bitów jednorazowo (najpierw LSB) (roz. 6.2 [1]):

- [SDI] bit 0 - bit RW:
  - 0: bity 8-15 są zapisywane do urządzenia,
  - 1: urządzenie podaje dane na linii SDO,
- [SDI] bity 1-7 - adres rejestru do odczytu lub zapisu (najpierw MSB),
- [SDI/SDO] bity 8-15 - dane transmitowane (pierwszy blok) (najpierw MSB),

- [SDI/SD0] bity od  $8 \cdot n$  do  $8 \cdot (n+1) - 1$  - dodatkowe bloki z danymi, gdzie  $n$  to numer dodatkowego bloku.

Transmisja jest zakończona po ustawieniu bitu **CS** na stan wysoki po min. 20 ns od ostatniego rosnącego zbocza zegara.

Poprawność komunikacji z urządzeniem sprawdzamy odczytując z niego rejestr **WHO\_AM\_I** pod adresem **0Fh**. Powinniśmy otrzymać wartość **69h**. (roz. 9.11 [1])

### 5.1.2 Opis odczytu danych z urządzenia pomiarowego

Po włączeniu urządzenia procedura uruchomieniowa trwa ok. 20 ms, potem akcelerometr i żyroskop automatycznie przechodzą w tryb "Power-Down".

Aby wybudzić akcelerometr i/lub żyroskop należy ustawić im częstotliwość aktualizacji danych. Ustawiamy więc 416 Hz dla akcelerometru ustawiając bity

**ODR\_XL[3:0] = 0110b(60h)** w rejestrze **CTRL1\_XL**. Dla żyroskopu (też 416 Hz) będą to bity **ODR\_G[3:0] = 0110b(60h)** w rejestrze **CTRL2\_G** (roz. 4.1 [2], roz. 9.12 [1], roz. 9.13 [1]).

Domyślnie urządzenie pomiarowe aktualizuje dane dla każdej osi pomiaru z akcelerometru/żyroskopu. Z racji, że nie używamy osi Z w żyroskopie zmieniamy bit **Zen\_G** na 0 w rejestrze **CTRL10\_C** (roz. 9.21 [1]).

Pomiary akcelerometru/żyroskopu znajdują się w zestawie rejestrów (adresy od **22h** do **2Dh**):

**OUT[XYZ]\_[LH]\_(G|XL),**

gdzie: **XYZ** - oś (wg rys. 1. [1]), **L** – rejestr z mniej znaczącym bajtem, **H** – rejestr z bardziej znaczącym bajtem, **G** – rejestr z danymi z żyroskopu, **XL** – rejestr z danym z akcelerometru (roz. 9.28 - 9.39).

Wartości znajdujące się w tych rejestrach interpretuje się następująco (roz. 4.5 [2]):

1. Bardziej znaczący bajt i mniej znaczący bajt dla danej osi i urządzenia (akcelerometr lub żyroskop) łączymy ze sobą do postaci słowa 16-bitowego:

$$OUT = (OUT\_H \ll 8) | OUT\_L.$$

2. Powstałe słowo **OUT** mnożymy przez wartość ilorazu: maksymalnej wartości zakresu pomiarowego, który jest aktualnie ustawiony w rejestrze **CTRL1\_XL** w bitach **FS\_XL[1:0]** dla akcelerometru (roz. 9.12 [1]) i **CTRL2\_G** w bitach **FS\_G[1:0]** dla żyroskopu (roz. 9.13 [1]), z liczbą dyskretnych wartości możliwych do zapisania w rejestrze dla wartości dodatnich, czyli  $2^{15}$  (połowa  $2^{16}$ , gdyż wartość w **OUT** jest ze znakiem (dopełnienie do dwóch)):



$$RES = OUT * \frac{\text{maks.wart.zak.pom.}}{2^{15}} [g \text{ lub } \frac{^{\circ}}{s}].$$

3. Otrzymany wynik *RES* jest w jednostce *g* (wielokrotność przyspieszenia ziemskiego ~ 9,81 m/s<sup>2</sup>) dla akcelerometru lub w jednostce °/s (stopnie na sekundę) dla żyroskopu.

Powyższe rejestry można odczytać w jednej, wielobajtowej transmisji SPI. W tym celu ustawiamy bit **IF\_INC** w rejestrze **CTRL3\_C** na 1. Od tego momentu urządzenie pomiarowe nie tylko zwróci wartość rejestru o żądanym adresie, ale też te o kolejnych adresach aż do momentu zmiany stanu bitu **CS** interfejsu SPI na wysoki (czyli do zakończenia transmisji) (roz. 6.2 [1]). Wykorzystujemy tę funkcję, aby odczytać rejestry **OUT...** (adresy od **22h** do **2Dh**).

### 5.1.3 Wbudowane funkcje (krokomierz) (roz. 6 [2])

Funkcje wbudowane w urządzenie pomiarowe LSM6DS3 do swojego działania używają akcelerometru. Działają one z częstotliwością 26 Hz.

Aby włączyć funkcję krokomierza należy:

1. Włączyć akcelerometr (roz. 5.1.2).
2. Włączyć wbudowane funkcjonalności – rejestr **CTRL10\_C**, bit **FUNC\_EN** na 1.
3. Włączyć algorytm krokomierza – rejestr **TAP\_CFG**, bit **PEDO\_EN** na 1.

Od tego momentu w rejestrach **STEP\_COUNTER\_H** i **STEP\_COUNTER\_L** będą pojawiać się wartości ilości kroków wykrytych od włączenia algorytmu krokomierza lub jego resetu (bit **PEDO\_RST\_STEP** ustawiony na 1 dla rejestru **CTRL10\_C**), które należy interpretować następująco:

$$\text{Ilość kroków} = (STEP\_COUNTER\_H \ll 8) | STEP\_COUNTER\_L.$$

Funkcja „Significant motion” (detektor zmiany lokalizacji użytkownika) potrzebuje do działania włączonego krokomierza, gdyż reaguje na konfigurowalną ilość wykrytych kroków. Zatem, procedura włączenia tej funkcji wygląda następująco:

1. Zmiana wartości progu (threshold), czyli liczby kroków w rejestrze **SM\_THS**. Jednak, aby móc zmienić ten rejestr, najpierw trzeba włączyć dostęp do rejestrów funkcji wbudowanych aktualizując rejestr **FUNC\_CFG\_ADDRESS** wartością **80h** (MSB tego rejestru na 1).
2. Procedura włączenia krokomierza jw., ale w pkt 4. ustawiając bit **INT1\_SIGN\_MOT** na 1 w rejestrze **INT1\_CTRL**.

## 5.2 Magnetometr AK09918

Dokumentacja urządzenia pomiarowego:

[3] [AK09918](#)

### 5.2.1 Opis interfejsu komunikacyjnego I2C

I2C do transmisji wykorzystuje dwie dwukierunkowe linie **SDA** – linię danych, oraz **SCL** – linię zegara. Obydwie na stałe podłączone do źródła zasilania przez rezystory podciągające pull-up. Magistrala I2C używa logiki dodatniej, czyli stan niski to logiczne 0 natomiast stan wysoki to logiczne 1. Podczas transmisji danych sygnał na linii **SDA** nie może się zmieniać, gdy linia **SCL** jest w stanie wysokim. Zmiany na linii **SDA** podczas stanu wysokiego na linii **SCL** są interpretowane jako sygnały kontrolne (startu, stopu). (roz. 10.1.1.1, 10.1.1.2 [3]).

- Start – utrzymujący się stan wysoki na linii **SCL**, zmiana stanu z wysokiego na niski na linii **SDA**
- Stop – utrzymujący się stan wysoki na linii **SCL**, zmiana stanu z niskiego na wysoki na linii **SDA**,
- Poprawność danych – po otrzymaniu warunku startu, stan linii **SDA** pozostaje niezmienny przy stanie wysokim linii **SCL**. Zmiana danych na linii **SDA** może nastąpić podczas stanu niskiego na linii **SCL**.
- **ACK/NACK** – po przesłaniu każdej porcji (czyli 8 bitów) danych, nadawca zwalnia linię **SDA**. Odbiorca w zależności czy walidacja przebiegła pomyślnie, podczas następnego impulsu zegara ustala stan na linii **SDA** na niski, wysyłając tym samym **ACK**, lub przeciwnie – wysoki, wysyłając **NACK**.

Każda porcja informacji wysyłana linią **SDA** ma długość jednego bajta. Każdy wysyłany bajt musi zostać potwierdzony przez odbiornik. Bity wysyłane są według zasady „bardziej znaczący bit jako pierwszy”. Jeśli urządzenie „slave” nie może w danej chwili wysłać lub odebrać danych transmisyjnych, może zasygnalizować to „masterowi” przez wstrzymanie linii **SCL** w stanie niskim. Transmisja zostanie wznowiona po zwolnieniu sygnału **SCL** przez „slave”.

## Obsługa AK09918 przez I2C:

Wspierane są dwa tryby czasowe:

- Standard mode –  $f_{SCL} \leq 100\text{kHz}$ ,
- Fast mode –  $100\text{kHz} \leq f_{SCL} \leq 400\text{kHz}$ .

(roz. 10.1 [3])

Sygnał zegarowy generowany jest przez urządzenie nadrzędne – „master”.

Zawartość ramki adresowej:

- 7-bitowy adres urządzenia „slave” = `0Ch` (unikalny, identyfikujący urządzenie w „I2C bus”),
- bit `READ/WRITE` stanowiący informację o tym czy mamy do czynienia z trybem „read” (1) , czy też „write” (0).

W rezultacie wysyłane ramki adresowe dla AK09918 mogą przyjąć formaty

8-bitowe: `0x19` - odczyt, `0x18` – zapis. (rys. 10.4 oraz roz. 10.1.1.4 [3])

Adresy dostępne podczas transmisji danych z/do AK09918: `00h` – `18h` oraz `30h` – `32h` (roz. 10.1.2, 10.1.3 [3]), przy zapisie wykorzystywane są tylko te z oznaczeniem „read/write” (roz 11.1 [3]). Dostępne adresy podlegają inkrementacji zgodnie z podanymi wyżej zakresami. Po adresie `18h` następuje powrót do `00h`.

Tryb zapisu (roz 10.1.2 [3]):

- 1) „Master” ustala warunek `START` oraz wysyła ramkę adresową z bitem `R/W` = 0 do wyznaczonego urządzenia „slave”, odbiorca potwierdza poprawność otrzymanych danych przez `ACK`.
- 2) „Master” wysyła adres rejestru wewnętrznego, następuje walidacja identyczna jak w punkcie 1).
- 3) Następuje transmisja danych przebiegająca zgodnie z podanymi w punkcie 5.2.1 ogólnymi założeniami, która kończy się przy rozpoznaniu warunku `STOP`.

Tryb odczytu (roz 10.1.3 [3]):

- sekwencyjny:

- 1) „Master” ustala warunek `START` oraz wysyła ramkę adresową z bitem `R/W` = 1 do wyznaczonego urządzenia „slave”, odbiorca potwierdza poprawność otrzymanych danych przez `ACK`.
- 2) Następuje transmisja danych przebiegająca zgodnie z podanymi w punkcie 5.2.1 ogólnymi założeniami, podczas której następuje sekwencyjny odczyt danych z dostępnych rejestrów.
- 3) Transmisja kończy się po otrzymaniu warunku stopu zamiast potwierdzenia po transmisji bajta danych.

- z wybranego adresu:

Początkowe postępowanie jest takie samo jak przy trybie zapisu, zatem punkty 1) oraz 2) pokrywają się z wcześniejszymi założeniami.

3) Po otrzymaniu **ACK** master ponownie wysyła ramkę adresową, tym razem ustalając bit **R/W = 1**. Odbiorca potwierdza poprawność wysyłając **ACK**.

4) Następuje transmisja danych z wybranego adresu, która przechodzi następnie w sekwencyjny odczyt danych. Transmisja kończy się po rozpoznaniu warunku **STOP**.

## 5.2.2 Tryby pracy czujnika

AK09918 posiada 7 różnych trybów pracy:

- (a) power-down mode,
- (b) single measurement mode,
- (c) continuous measurement mode 1 – 4,
- (d) self-test mode.

Wybór danego trybu jest możliwy przez odpowiednie ustawienie bitów rejestru **CNTL2 (0x31)** na bitach **[0:4]**. Po resecie mamy do czynienia ze stanem 0 na każdym bicie rejestru **31h**, zgodnie z dokumentacją jest to „power-down mode”. (roz. 11.3.8 [3])

Opis poszczególnych trybów pracy:

- W trybie „power-down” wszystkie rejestry wraz z zapisanymi w nich danymi są dostępne, mogą zostać zresetowane przez „soft reset” z odpowiednią opcją. (roz. 11.3.9 [3])  
„Soft reset” posiada dwie opcje „normal” oraz „reset”. Przy ustawieniu **1sb = 1**, w rejestrze **CNTL3 (0x32)** następuje ponowna inicjalizacja wszystkich rejestrów, a następnie bit **SRST** (lsb rejestru **CNTL3**) jest ponownie ustawiany jako „0”.
- Wykonanie jednego pomiaru i powrót do stanu „power-down”.
- Cykliczne wykonywanie pomiarów, zgodnie z dostępnymi wariantami częstotliwości ich dokonywania udostępnionymi dla sensora, czyli: 10Hz, 20Hz, 50Hz, 100Hz. Aby dokonać wyboru konkretnego wariantu, bity **[0:4]** rejestru **CNTL2 (0x31)** muszą znajdować się odpowiednio w stanach: „00001”, „00010”, „00100”, „00110”, „01000”. Aby zakończyć cykliczny pomiar należy przejść do trybu „power-down”.

Przy zmianie aktualnie wykonywanego trybu urządzenie przechodzi najpierw do stanu „power-down”, a następnie zmienia tryb na wybrany przez użytkownika. (czas pomiędzy zmianą ze stanu „power-down”, a wybranym trybem to minimalnie 100 µs)

### 5.2.3 Opis odczytu danych z urządzenia pomiarowego

Dane pomiarowe dotyczą trzech różnych wartości: indukcji magnetycznej w osi X, osi Y oraz osi Z. Są one zapisywane w rejestrach HXL – HZH w postaci surowej (roz. 11.3.4 [3]). Dla wartości indukcji magnetycznej osi X: rejestry 8-bitowe HXL (0x11) oraz HXH (0x12), według osi Y: rejestry 8-bitowe HYL (0x13) oraz HYH (0x14), dla osi Z: rejestry 8 bitowe HZL (0x15) oraz HZH (0x16).

Przykładowo, dla osi X: bity rejestru HXL jako „niższa” część otrzymywanego w wyniku słowa (16 bitów) powinny zająć pozycje [0:7] rejestru wynikowego, zatem:

$$(HXH \ll 8) | HXL.$$

Wynik uzyskiwany jest jako 16-bitowa liczba w kodzie U2.

Czujnik posiada limit zakresu mierzonych wartości, suma wartości indukcji magnetycznej dla wszystkich osi pomiarowych musi spełniać warunek:

$$|X| + |Y| + |Z| < 4912 \mu T.$$

Jeśli następuje przekroczenie zakresu bit D3 (H0FL) rejestru ST2 (0x18) = 1. (roz. 11.3.6. [3]). Wartości możliwe do odczytania z otrzymanego 16-bitowego rejestru wynikowego są z zakresu -32752 (-4912  $\mu T$  min) do 32752 (4912  $\mu T$  max).

Przetworzenie wynikowego słowa w celu otrzymania wartości indukcji magnetycznej dla mierzonej osi:

$$((HXH \ll 8) | HXL) * 0.15 [\mu T].$$

## 5.3 Czytnik kart NFC PN544

Skrócona dokumentacja urządzenia pomiarowego:

[4] [PN544 Objective Short data sheet](#)

Podręcznik użytkownika urządzenia pomiarowego:

[5] [PN544 C2 User Manual](#)

Interfejs kontrolera hosta (HCI- Host Controller Interface):

[6] [ETSI TS 102 622 V7.5.0](#)

### 5.3.1 Opis interfejsu komunikacyjnego UART

UART w odróżnieniu od I2C bądź SPI nie jest jedynie protokołem komunikacji, ale również fizycznym układem, jego głównym zadaniem jest szeregową transmisję i odbiór danych, natomiast sam układ może odbierać dane równolegle. Podczas komunikacji między układami dane przesyłane są bit po bicie. Jeden układ UART może zostać połączony tylko z jednym (innym) układem. PN544 oraz SOC kirin970 wykorzystują interfejs UART w wersji „High Speed Universal Asynchronous Receiver-Transmitter”.

Na linii **PVDD** układu PN544 podawane jest napięcie  $1.62 V_{min} < 3.0 V_{typ} < 3.3 V_{max}$ .

Podczas uruchomienia urządzenia (lub jego restartu) sprawdzana jest wartość na **IFSEL[2:0]** odpowiednio piny **F5 (IFSEL2)**, **F4 (IFSEL1)**, **A1 (IFSEL0)**, dla interfejsu UART na powyższych pinach podawane jest logiczne 0.

W celu obustronnej komunikacji układy korzystają z dwóch jednokierunkowych linii **Rx** (odbiorczej) oraz **Tx** (nadawczej), dodatkowo wykonywane jest krosowanie linii **Rx** i **Tx** układów. Dla układu znajdującego się po stronie *soc-kirin970* pin **Rx** zostaje połączony z **IF3** (pin **D4**) **Tx** układu PN544, pin *soc-kirin970* **Tx** zostaje połączony z **IF1** (pin **B4**) **Rx** układu PN544. Pozostałe piny danych PN544 **IF0** (pin **E4**) oraz **IF2** (pin **C4**) zostają rozłączone (roz. 6.1 [5])(roz.7 [4]).

Protokół komunikacyjny w warstwie fizycznej jest podobny do RS232 (bez sygnałów np. DTR, DSR, RI, Automatyczna negocjacja). UART jest interfejsem asynchronicznym przez co, nie posiada on linii pozwalającej synchronizować zegar między układami, zamiast tego należy ustawić szybkość transmisji, dla obu układów, dla PN544 wynosi ona 115200 bodów (boud rate), natomiast dozwolone są minimalne odchylenia od tej szybkości.

Dane między UART przesyłane są za pomocą ramek w warstwie fizycznej, dla PN544 wykorzystywany jest tryb „full duplex” z ośmioma bitami danych, brakiem bitu parzystości, jednym bitem stopu, jednym bitem startu, szybkością 115200 bodów oraz kolejnością danych: najmniej znaczący bit jako pierwszy.

Bit startu posiada wartość logiczną 0 (od +3V) natomiast bit stopu 1 (od -3V).

### 5.3.2 Logiczna warstwa łącza (Logic Layer Link / LLC)

Protokół komunikacji dla PN544 w warstwie logicznej został zapożyczony ze specyfikacji ETSI SWP, nie zostały zaimplementowane:

- Protokół CLT,
- Protokół ACT i obsługa SYNC\_ID.

Format ramki znajduje się w roz. 7.4.1 [5]

Podczas komunikacji wymagane jest co najmniej 1.1ms opóźnienia między kolejnymi ramkami. Dokładny opis struktury komend HCI znajduje się w roz. 7.4 [5].

Przed zainicjowaniem interfejsu komunikacji (przy uruchomieniu lub resecie [pin VEN]), konieczne jest wysłanie ramki RSET (roz. 7.4.3.3 [5]), aby zresetować status połączenia.

Suma kontrolna CRC-16 liczona jest z wartości początkowej 0xFFFF wielomianem  $x^{16} + x^{12} + x^5 + 1$ , pomiędzy bitami SOF i EOF z ich wyłączeniem. Poprawnie wysłane ramki zwracają ACK. Do transmisji danych używany jest strumień bajtów, bez potwierdzeń między znakami. W przypadku wystąpienia błędu następuje retransmisja ramki.

### 5.3.3 Zarządzenie PN544

Korzystając z UART nie używamy zegara więc pod adres 0x9E71 (PlClockRequest) wpisujemy wartość 0x00. Do komunikacji wykorzystane zostają karty o standardzie ISO14443-4 (NXP MIFARE Plus). W trybie odczytu dla ISO14443 opóźnienie aktywacji musi być mniejsze niż wartości znajdujące się pod adresem 0x9F2C (TGINIT\_GUARD\_TO\_MSB) domyślnie 0x51 oraz 0x9F2D (THINIT\_GUARD\_TO\_LSB) domyślnie 0x61 - opóźnienie aktywacji od 0 do 3.145s z 48 mikro-sekundowymi odstępami. Następnie należy wybrać które typy tagów mają zostać obsługiwane, w tym celu do rejestru NXP\_PL\_RDPHAES wpisujemy wartości wg roz. 9.6.2.4 [5], kolejno przechodzimy w tryb łączenia- w rejestrze NXP\_PL\_PAUSE znajduje się wartość 0x0824 (100ms), określa długość trwania fazy zatrzymania skanowania- proces skanowania znajduje się na Fig. 51 str. 88 [5], natomiast proces łączenia z kartą ISO14443 znajduje się w roz. 9.6.3 [5]. Faza odczytu oczekuje na wywołanie zdarzenia EVT\_TARGET\_DISCOVERED przez PN544, przesyłamy w nim informację o ilości wykrytych tagów, wartość 0x00 informuje o pojedynczym tagu, natomiast wartość 0x03 powiadamia, że znajduje się kilka tagów oraz proces odczytu powinien zostać przerwany roz. 10.3.4.1 [6]. Następnie host odpowiada komendą WR\_XCHG\_DATA z CTR (1 bajt) zawierającym na bitach [4:1] wartość x, które znajduje się w przedziale <0; 14>, wpływają one na opóźnienie, które wyliczamy ze wzoru:

$$RES = \left( \frac{256 \cdot 16}{13.56 \text{ MHz}} \right) * 2^x.$$

Kolejno PN544 komunikuje się protokołem ISO14443 z tagiem (kartą) pobierając z niego dane oraz zwraca `ANY_OK` do hosta zawierający otrzymane dane oraz wskaźnik błędu RF (0 - brak błędu; 1 - błąd). Opis rejestrów bramki RF jest opisany w roz. 10.2.3.1 [6]. W celu zakończenia wymiany informacji host wysyła zdarzenie `EVT_END_OPERATION`, aby rozpocząć kolejne skanowanie tagów- host musi wysłać zdarzenie `EVT_READER_REQUESTED`.