**Global Logic**

**Протокол комунікації комп’ютера з інтегральною мікросхемою**

Для подальшої роботи з проектом ми продумали правила за якими комп’ютер та мікросхема будуть обмінюватись даними. Обмін даними проводиться з метою зробити пристрій, що аналізує запит який може обробити та дати певну задану відповіть . Задумано так, дані з компютерної програми будуть висилатись на мікроконтролер та аналізуватимуться та на їх основі мікрокомп’ютер буде давати поради чіткі відповіді на задані нами компанди .

По-перше, щоб спілкуватись одних даних недостатньо. Для успішної комунікації їх потрібно загорнути в зрозумілу форму - як для отримувача, так і для відправника. Саме тому комунікацію виду “Request-Response” можна описати декількома UART

Спочатку розглянемо запити, що відправляються з комп’ютера, а пізніше буде описані запити мікроконтролера.

1. Запит { Is device connected ? } на контроллер будуть надіслані на задану відповідь тільки так

|  |  |
| --- | --- |
| Command  1 Байт | Chakesum  1 Байт |
| 0x01 | CRC |

Request

Response

|  |  |  |
| --- | --- | --- |
| Command  1 Байт | Confirmation  1 Байт | Chakesum  1 Байт |
| 0x01 | 0x01 yes | CRC |

0x01 - команда 0 байт

CRC надлишковий залишок коду

1. Запит { Can you generate kay ? } запит з компа на мінкроконртолер чи достатньо йому доаних що він почав генерувати ключ .

request

|  |  |
| --- | --- |
| Command  1 Байт | Chakesum  1 Байт |
| 0x02 | CRC |

Responce

|  |  |  |
| --- | --- | --- |
| Command  1 Байт | Confirmation  1 Байт | Chakesum  1 Байт |
| 0x02 | 0x01 (yes)\ 0x00(no) | CRC |

0x01 (yes)\ 0x00(no) CRC після такого як проограма зрозуміла що ій достатньо данних відповіддає так або ні .

CRC надлишковий залишок коду

1. Запит { Generate kay } після того як програма зрозуміла чи данних було достатньо та помилок не було : відповіла (yes) та згенерувала ключ

request

|  |  |
| --- | --- |
| Command  1 Байт | Chakesum  1 Байт |
| 0x03 | CRC |

responce

|  |  |  |
| --- | --- | --- |
| Command  1 Байт | Confirmation  1 Байт | Chakesum  1 Байт |
| 0x03 | 0x03 (yes) | CRC |

0x03 (yes) команда генерування ключа має мати відповідь тільки так або буде вибивати що бракує даних для генерування

CRC надлишковий залишок коду

1. Запит { Encrypy data } запит який дає команду зашифрувати дані та отрримати розшифровані дані або вибити помилку якщо є якісь трабли

request

|  |  |  |
| --- | --- | --- |
| Command  1 Байт | Confirmation  1 Байт | Chakesum  1 Байт |
| 0x04 | (data size =4 bytes ) (data)CRC | CRC |

responce

|  |  |  |  |
| --- | --- | --- | --- |
| Command  1 Байт | Confirmation  1 Байт | Confirmation  1 Байт | Chakesum  1 Байт |
| 0x04 | data size =4 bytes ) (encrypt data) | 0x00n 0x00 0x00 0x00 (error code = 1 byte ) | CRC |

Max size bytes 512

0 байт 0х00

CRC надлишковий залишок коду

1. Запит {Decrypt data } запит який дає команду розшифрувати дані та отрримати зашифровані дані або вибити помилку якщо є якісь трабли.

request

|  |  |  |
| --- | --- | --- |
| Command  1 Байт | Confirmation  1 Байт | Chakesum  1 Байт |
| 0x05 | (Data size =4bytes) (data) | CRC |

responce

|  |  |  |  |
| --- | --- | --- | --- |
| Command  1 Байт | Confirmation  1 Байт | Confirmation  1 Байт | Chakesum  1 Байт |
| 0x05 | data size = 4 bytes)(decrypted data) | 0x00 0x00 0x00 0x00 (error code =1 byte) | CRC |

0 - байт 0x00

CRC надлишковий залишок коду

Кидаємо ми дані не по чотири байти, а по стільки скільки вкажемо у data size. Тобто ми виділяємо 4 байти на те, щоб вказати data size, а в тих 4ох байтах ми можемо вказати будь-яке число від 1 до 512. Тобто, якщо ми вкажемо 512 у місці для data size (на яке ми виділили 4 байти), то це означатиме що дані, які ми надсилаємо займатимуть 512 байт.