Requirements and Detailed Design
Serial Transmission

Brief:
An IoT system that consists of a Raspberry Pi and a micro-controller communicating to each other via Serial Communication.
The data is sent:
uint8_t type;
int8_t rssi;
uint16_t length; // Length of wifi_data array
int8_t wifi_data; // Dynamic array of CSI data*
uint32_t timestamp;

Requirements

1. Create a C code
   1.1 Create a struct with the given data seen in the brief
   1.2 Write functions to construct the payload in binary format
   1.3 Send the data using serial communication.

2. Make a sample C code which fills the struct with random values and stream via serial communication.

3. Use a low level UART C function

4. Create a Python code to get the serial data and extract the values and print the values.

Detailed Design for C code:

DD-1:

Covered requirement: 1.1. requirement - Create a struct with the given data

Brief: create a struct with the following data:

uint8_t type;
int8_t rssi;
uint16_t length; // Length of wifi_data array
int8_t wifi_data; // Dynamic array of CSI data*
uint32_t timestamp;

DD-2:

Covered requirement: 2. requirement - Make a sample C code which fills the struct with random values and stream via serial communication.

Brief: This function fills the WifiPacket struct with random values.

DD-3:

Brief: Create and use CRC as an error-detecting element

DD-4:

Covered requirement: 1.2. requirement - Write functions to construct the payload in binary format

Brief: This function serializes the WifiPacket, allocates memory

DD-4.1 - Serializate the WifiPacket
DD-4.2 – Calculate payload size
DD-4.3 – Allocate memory – returns NULL if allocation fails – **Error handling**

DD-5:

Covered requirement: 2. requirement - Send the data using serial communication

Brief: This function checks UART availability, write bytes and close connection

DD-5.1 - function : Write to UART– use default port: „serial0”
DD-5.2 – Check UART availability– **Error handling**
DD-5.3 – Write to UART and close connection

DD-6:

Covered requirement: 2. requirement - Send the data using serial communication

Brief: This function sends packets and gives feedback in hexadecimal format to support debugging and monitoring.

DD-7:

Covered requirement: 1.2. requirement - Write functions to construct the payload in binary format

Brief: Free memory

Detailed Design for Python code:

Covered requirement: 4. requirement - Create a Python code to get the serial data and extract the values and print the values.

Explanation can be found in the code due to time restrictions.