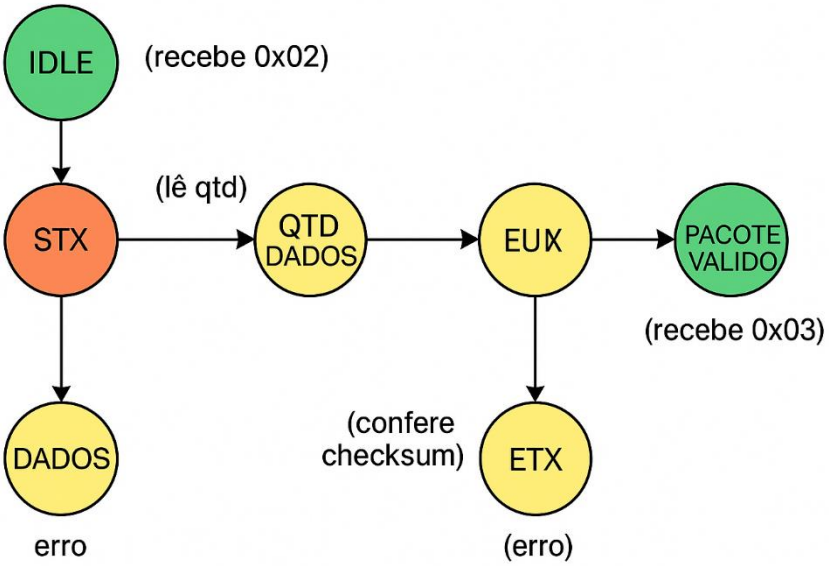
PROJETO DE SISTEMAS EMBARCADOS

26/08/2025

RAVI DE FARIAS

DIAGRAMA DE FUNCIONAMENTO DO SISTEMA:



CODIGO RECEPTOR:

```c
#include <stdio.h>
#include <stdint.h>

#define STX 0x02
#define ETX 0x03

typedef enum {
    ST_IDLE,
    ST_STX,
    ST_QTD,
    ST_DADOS,
    ST_CHK,
    ST_ETX,
    ST_OK,
    ST_ERROR
} State;

static State fsm_state = ST_IDLE;

static uint8_t buffer[256];
static uint8_t qtd_dados = 0;
static uint8_t dados_count = 0;
static uint8_t checksum_calc = 0;
static uint8_t checksum_rx = 0;

void reset_fsm() {
    fsm_state = ST_IDLE;
    qtd_dados = 0;
```

```
Transmitindo bytes...


...Program finished with exit code 0
Press ENTER to exit console.
```

```c
void reset_fsm() {
    fsm_state = ST_IDLE;
    qtd_dados = 0;
    dados_count = 0;
    checksum_calc = 0;
    checksum_rx = 0;
}

void fsm_receiver(uint8_t byte) {
    switch(fsm_state) {
        case ST_IDLE:
            if (byte == STX) {
                fsm_state = ST_STX;
            }
            break;

        case ST_STX:
            qtd_dados = byte;
            dados_count = 0;
            checksum_calc = 0;
            fsm_state = ST_QTD;
            break;

        case ST_QTD:
            buffer[dados_count++] = byte;
            checksum_calc ^= byte;
```

```
nsmitindo bytes...


Program finished with exit code 0
ss ENTER to exit console.
```

```
        case ST_QTD:
            buffer[dados_count++] = byte;
            checksum_calc ^= byte;
            if (dados_count >= qtd_dados) {
                fsm_state = ST_DADOS;
            }
            break;

        case ST_DADOS:
            checksum_rx = byte;
            if (checksum_rx == checksum_calc) {
                fsm_state = ST_CHK;
            } else {
                fsm_state = ST_ERROR;
            }
            break;

        case ST_CHK:
            if (byte == ETX) {
                fsm_state = ST_OK;
            } else {
                fsm_state = ST_ERROR;
            }
            break;

        case ST_OK:
            printf("Pacote recebido com sucesso! Dados: ");
```

```
ir
nsmitindo bytes...


Program finished with exit code 0
ss ENTER to exit console.
```

```
            }
            break;

        case ST_OK:
            printf("Pacote recebido com sucesso! Dados: ");
            for (int i = 0; i < qtd_dados; i++) {
                printf("%02X ", buffer[i]);
            }
            printf("\n");
            reset_fsm();
            break;

        case ST_ERROR:
            printf("Erro no pacote!\n");
            reset_fsm();
            break;
    }
}

int main() {
    uint8_t pacote[] = {0x02, 0x03, 0x10, 0x20, 0x30, (0x10^0x20^0x30), 0x03};

    printf("Transmitindo bytes...\n");
    for (int i = 0; i < sizeof(pacote); i++) {
        fsm_receiver(pacote[i]);
    }
```

```
input
nsmitindo bytes...


Program finished with exit code 0
ss ENTER to exit console.
```

CÓDIGO TRANSMISSOR:

```c
#include <stdio.h>
#include <stdint.h>

#define STX 0x02
#define ETX 0x03

uint8_t calcula_checksum(uint8_t *dados, uint8_t qtd) {
    uint8_t chk = 0;
    for (int i = 0; i < qtd; i++) {
        chk ^= dados[i];
    }
    return chk;
}

void transmissor_send(uint8_t *dados, uint8_t qtd) {
    uint8_t chk = calcula_checksum(dados, qtd);

    printf("Enviando pacote:\n");

    printf("%02X ", STX);

    printf("%02X ", qtd);

    for (int i = 0; i < qtd; i++) {
        printf("%02X ", dados[i]);
    }

    printf("%02X ", chk);
```

```
Enviando pacote:
02 02 AB CD 66 03


...Program finished with exit code 0
Press ENTER to exit console.
```

```c
void transmissor_send(uint8_t *dados, uint8_t qtd) {
    uint8_t chk = calcula_checksum(dados, qtd);

    printf("Enviando pacote:\n");

    printf("%02X ", STX);

    printf("%02X ", qtd);

    for (int i = 0; i < qtd; i++) {
        printf("%02X ", dados[i]);
    }

    printf("%02X ", chk);

    printf("%02X\n", ETX);
}

int main() {
    uint8_t dados1[] = {0x10, 0x20, 0x30};
    uint8_t dados2[] = {0xAB, 0xCD};

    transmissor_send(dados1, sizeof(dados1));
    transmissor_send(dados2, sizeof(dados2));

    return 0;
}
```

```
nviando pacote:
2 02 AB CD 66 03


..Program finished with exit code 0
ress ENTER to exit console.
```

TRANSMISSOR + RECEPTOR:

```c
#include <stdio.h>
#include <stdint.h>

#define STX 0x02
#define ETX 0x03

/* ----------------------- RECEPTOR ----------------------- */
typedef enum {
    RX_IDLE,
    RX_WAIT_QTD,
    RX_READ_DATA,
    RX_WAIT_CHK,
    RX_WAIT_ETX
} RxState;

static RxState rx_state = RX_IDLE;

static uint8_t rx_buf[256];
static uint8_t rx_expected = 0;
static uint8_t rx_len = 0;
static uint8_t rx_chk = 0;

static void rx_reset(void) {
    rx_state = RX_IDLE;
    rx_expected = 0;
    rx_len = 0;
    rx_chk = 0;
}

static void rx_byte(uint8_t b) {
    switch (rx_state) {
    case RX_IDLE:
        if (b == STX) {
            rx_state = RX_WAIT_QTD;
        }
        break;

    case RX_WAIT_QTD:
        rx_expected = b;
        rx_len = 0;
        rx_chk = 0;
        rx_state = (rx_expected > 0) ? RX_READ_DATA : RX_WAIT_CHK;
        break;

    case RX_READ_DATA:
        if (rx_len < sizeof(rx_buf)) {
            rx_buf[rx_len++] = b;
            rx_chk ^= b;
        }
        if (rx_len >= rx_expected) {
            rx_state = RX_WAIT_CHK;
```

```
            rx_state = RX_WAIT_CHK;
        }
        break;

    case RX_WAIT_CHK:
        if (b == rx_chk) {
            rx_state = RX_WAIT_ETX;
        } else {
            printf("ERRO: checksum invalido\n");
            rx_reset();
        }
        break;

    case RX_WAIT_ETX:
        if (b == ETX) {
            printf("OK: pacote recebido. Dados: ");
            for (uint8_t i = 0; i < rx_expected; i++) {
                printf("%02X ", rx_buf[i]);
            }
            printf("\n");
        } else {
            printf("ERRO: ETX invalido\n");
        }
        rx_reset();
        break;
    }
}
```

```
        }
        rx_reset();
        break;
    }
}

/* ------------------------ TRANSMISSOR ------------------------ */
static uint8_t tx_calc_chk(const uint8_t *data, uint8_t n) {
    uint8_t c = 0;
    for (uint8_t i = 0; i < n; i++) {
        c ^= data[i];
    }
    return c;
}

static void tx_send(const uint8_t *data, uint8_t n) {
    uint8_t chk = tx_calc_chk(data, n);

    rx_byte(STX);
    rx_byte(n);
    for (uint8_t i = 0; i < n; i++) {
        rx_byte(data[i]);
    }
    rx_byte(chk);
    rx_byte(ETX);
}
```

```
    rx_byte(n);
    for (uint8_t i = 0; i < n; i++) {
        rx_byte(data[i]);
    }
    rx_byte(chk);
    rx_byte(ETX);
}

int main(void) {
    const uint8_t p1[] = {0x10, 0x20, 0x30};
    const uint8_t p2[] = {0xAB, 0xCD};

    tx_send(p1, (uint8_t)sizeof p1);
    tx_send(p2, (uint8_t)sizeof p2);

    const uint8_t bad[] = {0x01, 0x02, 0x03};
    uint8_t chk_bad = tx_calc_chk(bad, (uint8_t)sizeof bad) ^ 0xFF;
    rx_byte(STX);
    rx_byte((uint8_t)sizeof bad);
    for (uint8_t i = 0; i < (uint8_t)sizeof bad; i++) rx_byte(bad[i]);
    rx_byte(chk_bad);
    rx_byte(ETX);

    return 0;
}
```

input

```
OK: pacote recebido. Dados: 10 20 30
OK: pacote recebido. Dados: AB CD
ERRO: checksum invalido


...Program finished with exit code 0
Press ENTER to exit console.
```