

```

#include "mcc_generated_files/system/system.h"
#include "mcc_generated_files/uart/eusart1.h"
#include "mcc_generated_files/spi/mssp1.h"
#include "mcc_generated_files/spi/mssp2.h"
#include <xc.h>
#include <stdint.h>
#include <stdbool.h>
#include <stdio.h>

#define MSG_SIZE 5 // 5 byte messages
#define START 0x41 // Start byte of message frame
#define USER_ID 0x05 // This device's ID
#define END 0x42 // End byte of message frame

uint8_t uart_buffer[MSG_SIZE];
uint8_t index = 0;
bool receiving = false;

void run_motors_forward(void) {
    // Set all motor pins HIGH to activate both motors
    IO_RA7_SetHigh(); // Motor 1 +
    IO_RB4_SetLow(); // Motor 1 -

    IO_RC0_SetHigh(); // Motor 2 +
    IO_RC2_SetLow(); // Motor 2 -
}

void stop_motors(void) {
    // Set all motor control pins LOW to stop both motors
    IO_RA7_SetLow();
    IO_RB4_SetLow();
    IO_RC0_SetLow();
    IO_RC2_SetLow();
}

void send_message_raw(uint8_t *msg) {
    for (uint8_t i = 0; i < MSG_SIZE; i++) {
        EUSART1_Write(msg[i]);
    }
    printf("Message forwarded: %02X %02X %02X %02X %02X\n",
        msg[0], msg[1], msg[2], msg[3], msg[4]);
}

void send_message(uint8_t receiver_id, uint8_t data) {
    uint8_t msg[MSG_SIZE] = {START, USER_ID, receiver_id, data, END};

```

```

    for (int i = 0; i < MSG_SIZE; i++) {
        EUSART1_Write(msg[i]);
    }
    printf("Message sent: %02X %02X %02X %02X %02X\n",
        msg[0], msg[1], msg[2], msg[3], msg[4]);
}

void perform_action(uint8_t command) {
    switch (command) {
//      case 0x64:
//          IO_RA0_SetHigh();IO_RA1_SetLow(); IO_RA2_SetLow();IO_RA3_SetLow(); break; //
Left
        case 0x00:
            IO_RA0_SetLow();IO_RA1_SetLow();
            IO_RA2_SetHigh();IO_RA3_SetLow();stop_motors(); break; // Stop
//      case 0x65:
//          IO_RA0_SetLow();IO_RA1_SetHigh(); IO_RA2_SetLow();IO_RA3_SetLow(); break; //
Right
        case 0x01:
            IO_RA0_SetLow();IO_RA1_SetLow();
            IO_RA2_SetLow();IO_RA3_SetHigh();run_motors_forward(); break; // Forward
        default:
            IO_RA2_SetHigh(); IO_RA3_SetHigh(); break; // Error LEDs
    }
}

void process_message(uint8_t *msg) {
    if (msg[0] == START && msg[4] == END) {
        printf("Message Received: %02X %02X %02X %02X %02X\n",
            msg[0], msg[1], msg[2], msg[3], msg[4]);
        if (msg[2] == USER_ID) {
            perform_action(msg[3]);
        } else if (msg[1] != USER_ID) {
            send_message_raw(msg); // Forward as-is
        }
    } else {
        printf("Malformed message ignored.\n");
    }
}

// Check UART and receive message byte-by-byte
void check_uart_receive() {
    while (EUSART1_IsRxReady()) {
        uint8_t byte = EUSART1_Read();
        if (!receiving) {
            if (byte == START) {

```

```

        uart_buffer[0] = byte;
        index = 1;
        receiving = true;
    }
} else {
    uart_buffer[index++] = byte;
    if (index >= MSG_SIZE) {
        receiving = false;
        index = 0;
        if (uart_buffer[MSG_SIZE - 1] == END) {
            process_message(uart_buffer);
        } else {
            printf("Invalid end byte: %02X\n", uart_buffer[MSG_SIZE - 1]);
        }
    }
}
}
}
}
}

```

```

// RC7 and RB4 are a +- motor pair
// RC0 and RC2 are a +- motor pair

```

```

int main(void) {
    SYSTEM_Initialize();
    EUSART1_Initialize();
    SPI1_Initialize();
    SPI2_Initialize();

    IO_RA0_SetHigh();
    while (1) {
        check_uart_receive();
    }
}

```