```
#include <stdio.h>
#include <stdlib.h>

void extractPayload(const uint8_t *inputArray, size_t arrayLength) {
    // Check if the array length is at least 5 (command + length + data type)
    if (arrayLength < 5) {
        printf("Invalid input array length\n");
        return;
    }

    // Extract command, length, and data type
    uint8_t command = inputArray[0];
    uint16_t length = (inputArray[1] << 8) | inputArray[2];
    uint8_t dataType = inputArray[3];

    // Check if the array length is consistent with the specified length
    if (arrayLength != length + 4) {
        printf("Invalid input array length\n");
        return;
    }

    // Extract payload data
    uint8_t *payload = malloc(length - 1);
    if (payload == NULL) {
        printf("Memory allocation error\n");
        return;
    }

    for (size_t i = 0; i < length - 1; ++i) {
        payload[i] = inputArray[i + 4];
    }

    // Output the extracted payload data
    printf("Command: %02X\n", command);
    printf("Length: %04X\n", length);
    printf("Data Type: %02X\n", dataType);
    printf("Payload Data: ");
    for (size_t i = 0; i < length - 1; ++i) {
        printf("%02X ", payload[i]);
    }
    printf("\n");

    // Free allocated memory
    free(payload);
}

int main() {
    uint8_t inputArray[] = {0x00, 0x02, 0x00, 0x11, 0x01, 0x01, 0x02, 0x03, 0x04, 0x05, 0x06,
0x07, 0x08, 0x09, 0x10};
    size_t arrayLength = sizeof(inputArray) / sizeof(inputArray[0]);

    extractPayload(inputArray, arrayLength);

    return 0;
}
```