

# UB-E03詳細取扱説明書 Rev.B 誤記訂正

## 訂正箇所

- P24 Windowsサンプルプログラム
- P25 Linuxサンプルプログラム
- P26 印刷の権利

2015年7月10日  
セイコーエプソン株式会社

## 24ページ

Windowsサンプルプログラム

誤) プログラムが正しく動作しません

```
/* TCP9100 programming sample for win32
 * HOW TO BUILD
 *   cl wtcp.cpp wsock32.lib
 */
#include <stdio.h>
#include <winsock.h>

int main(int argc, char* argv[])
{
    WSADATA data;
    SOCKET sock;
    struct linger Linger;
    struct sockaddr_in addr;

    if (argc != 2) {
        printf("usage: wtcp <IP_ADDRESS>\n");
        exit(1);
    }

    /* Initialize windows socket */
    WSASStartup(0x0101, &data);

    /* Create socket */
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == INVALID_SOCKET) {
        fprintf(stderr, "Error socket(): %d\n", WSAGetLastError());
        exit(1);
    }

    /* Set connection timeout */
    Linger.l_onoff = 1;
    Linger.l_linger = 60;
    setsockopt(sock, SOL_SOCKET, SO_LINGER, (char*)&Linger, sizeof(struct linger));

    /* initialize the parameter */
    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_port = htons(9100);
    addr.sin_addr.s_addr = inet_addr(argv[1]);

    /* connect */
    if (connect(sock, (struct sockaddr*)&addr, sizeof(addr)) < 0) {
        fprintf(stderr, "Error connect(): %d\n", WSAGetLastError());
        exit(1);
    }
    printf("connected\n");

    /* send data */
    send(sock, "EPSON\n", 6, 0);

    /* gracefully close */
    shutdown(sock, 1); /* SD_SEND */
    while(1) {
        char buf[64];
        int n = recv(sock, buf, 64, 0);
        if(n == SOCKET_ERROR || n == 0)
            break;
    }
    shutdown(sock, 2); /* SD_BOTH */

    /* close socket */
    closesocket(sock);
    return 0;
}
```

# 24ページ

## Windowsサンプルプログラム

### 正) 修正版プログラム

```
/* TCP9100 programming sample for win32
 * HOW TO BUILD
 * cl wtcp.cpp wsock32.lib
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <winsock.h>

int main(int argc, char* argv[])
{
    WSADATA data;
    SOCKET sock;
    struct linger Linger;
    struct sockaddr_in addr;
    char buf[64];
    int n;

    if (argc != 2) {
        printf("usage: wtcp <IP_ADDRESS>\n");
        exit(1);
    }

    /* Initialize windows socket */
    WSASStartup(0x0101, &data);

    /* Create socket */
    if ((sock = socket(AF_INET, SOCK_STREAM, 0)) == INVALID_SOCKET) {
        fprintf(stderr, "Error socket(): %d\n", WSAGetLastError());
        exit(1);
    }

    /* Set connection timeout */
    Linger.l_onoff = 1;
    Linger.l_linger = 60;
    setsockopt(sock, SOL_SOCKET, SO_LINGER, (char*)&Linger, sizeof(struct
linger));

    /* initialize the parameter */
    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_port = htons(9100);
    addr.sin_addr.s_addr = inet_addr(argv[1]);

    /* connect */
    if (connect(sock, (struct sockaddr*)&addr, sizeof(addr)) < 0) {
        fprintf(stderr, "Error connect(): %d\n", WSAGetLastError());
        exit(1);
    }
    printf("connected\n");

    /* send data */
    send(sock, "EPSON UB-E03\n", 13, 0);

    /* gracefully close */
    shutdown(sock, 1); /* SD_SEND*/
    while(1) {
        n = recv(sock, buf, 64, 0);
        if (n == SOCKET_ERROR || n == 0) {
            break;
        }
    }
    shutdown(sock, 2); /*SD_BOTH*/

    /* close socket */
    closesocket(sock);
}
```

## 25ページ

Linuxサンプルプログラム

誤) プログラムが正しく動作しません

```
/* TCP9100 programming sample for LINUX
 * HOW TO BUILD
 *   cc ltcp.c
 */
#include <stdio.h>
#include <netdb.h>

int main(int argc, char* argv[])
{
    int sock;
    struct sockaddr_in addr;

    if (argc != 2) {
        printf("usage: ltcp <ip address>\n");
        exit(1);
    }

    /* create socket */
    sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock < 0){
        perror("socket()");
        exit(1);
    }

    /* initialize the parameter */
    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_port = htons(9100);
    addr.sin_addr.s_addr = inet_addr(argv[1]);

    /* connect */
    if (connect(sock, (struct sockaddr*)&addr, sizeof(addr)) < 0){
        perror("connect()");
    }
    printf("connected\n");

    /* send data */
    send(sock, "EPSON\n", 6, 0);

    /* close socket */
    close(sock);

    return 0;
}
```

# 25ページ

## Linuxサンプルプログラム

### 正) 修正版プログラム

```
/* TCP9100 programming sample for LINUX
 * HOW TO BUILD
 * cc ltcp.c
 */
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netdb.h>

int main(int argc, char* argv[])
{
    int sock;
    struct sockaddr_in addr;

    if (argc != 2) {
        printf("usage: ltcp <ip address>\n");
        exit(1);
    }

    /* create socket */
    sock = socket(AF_INET, SOCK_STREAM, 0);
    if (sock < 0) {
        perror("socket()");
        exit(1);
    }

    /* initialize the parameter */
    memset(&addr, 0, sizeof(addr));
    addr.sin_family = AF_INET;
    addr.sin_port = htons(9100);
    addr.sin_addr.s_addr = inet_addr(argv[1]);

    /* connect */
    if (connect(sock, (struct sockaddr*)&addr, sizeof(addr)) < 0) {
        perror("connect()");
    }
    printf("connected\n");

    /* send data */
    send(sock, "EPSON UB-E03\n", 13, 0);

    /* close socket */
    close(sock);
}
```

## 26ページ

誤) 最大3つまでの接続要求に対して接続を許可します。

### 印刷の権利

UB-E03 は、lpr/port9100 のプロトコルにかかわらず、最大3つまでの接続要求に対して接続を許可します。実際に TM プリンターに対して印刷が可能なのは、印刷権利を得られた最初の接続です。他の接続要求は、最初の接続がクローズ（ホストによる明示的なクローズ、またはタイムアウトによるクローズ）されるまで、データ送信がブロックされます。

正) 最大6つまでの接続要求に対して接続を許可します。