

NEC uPD78F9222 在线下载试验板

1. 引言

NEC uPD78F9222 是 78K0S/KA1+ 20MC 微控制器的初学者试验板。

1.1 NEC uPD78F9222 在线下载试验板的主要特征

- 易于使用的器件示范功能

NEC uPD78F9222 在线下载试验板中包含一些组件，可以很方便地演示简单的 LED 灯，七段代码管，蜂鸣器等 I/O 口操作。

- 通过 USB 接口供电

NEC uPD78F9222 在线下载试验板通过 USB 接口供电，不需要独立电源。

- WriteEZ2 闪存编程软件

可以通过基于闪存编程软件的窗口，选择和下载应用程序到 NEC uPD78F9222 在线下载试验板用于评测。

- 支持模拟信号到数字信号的转换
- 可以使用多种输入/输出信号

如：连接到用户硬件上的所有 I/O 端口、定时器输入/输出信号、使用 USB UART 芯片 FT232 的 UART 接口、连接到 LED 的 4 个 I/O 端口。

2. 软件安装

2.1 汇编程序和整合开发环境 PM+的安装

要安装包含有整合开发环境 PM+的汇编程序包，可以点击安装程序中 ra78k0s_w140_e.exe 启动 RA78K0S 安装程序。产品 ID: 00001083P

2.2 C 编译器的安装

要执行用 C 语言开发的程序，必须安装 C 编译器。点击安装程序中的 cc78k0s_w150_e 启动 CC78K0S 安装程序。

安装过程与 PM+的安装相同:ID 为 00001758P

2.3 系统仿真器的安装

点击安装程序中的 sm+for78k0s_kx1+_w102_e 启动 78K0S/Kx1+ 安装程序。

安装过程与 PM+的安装相同:ID 为 00001664C

2.5 GUI 软件驱动的安装

USB 驱动程序的安装

在连接实验板到电脑前，当 Windows 发现新的硬体后，把刚才解压的驱动程式路径输入。Windows 会自行安装实验板的 USB 驱动程式。请使用光盘 Drivers 文件夹下的驱动程序。

3. 整合开发环境 PM+和系统仿真器 SM+

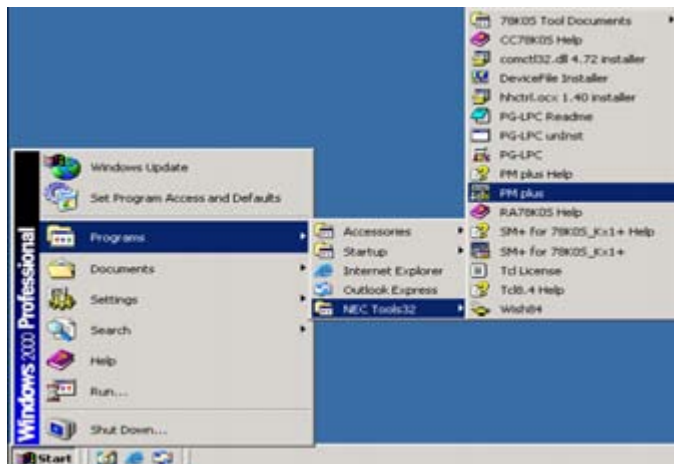
通过执行简单的程序，讲述用于 78K0S（以后用 SM+代替）的整合开发环境 PM+和系统仿真器 SM+的基本操作。这章中假设的环境如下列所示。

所用的示例名: seven1

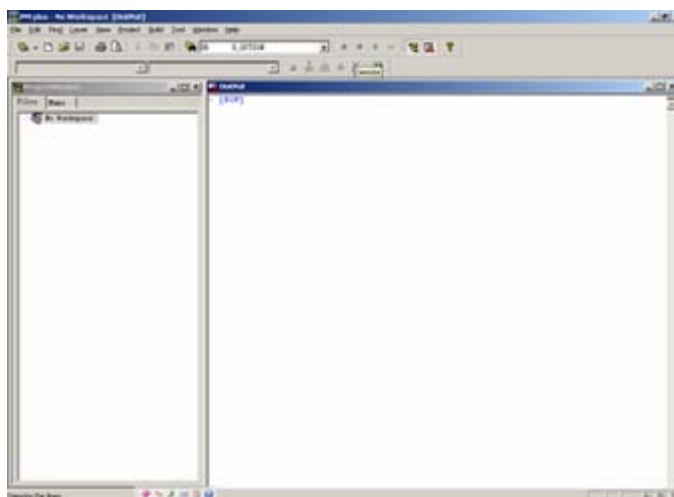
所用 workspace: sevendemo.prw

3.1 PM+的启动和编译

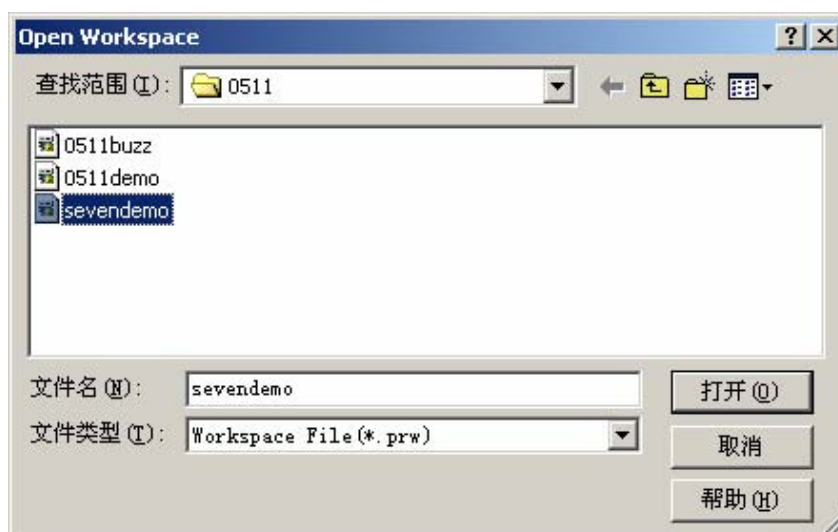
1. 启动 PM+。在正常的安装方法下，通过顺序点击 开始 - 程序 - NECTools32 - PMplus 可以启动 PM+。



2. 当 PM+ 启动，显示如图所示窗口。在折叠试菜单中点击 File - Open Workspace ，并打开工作区选择对话框。

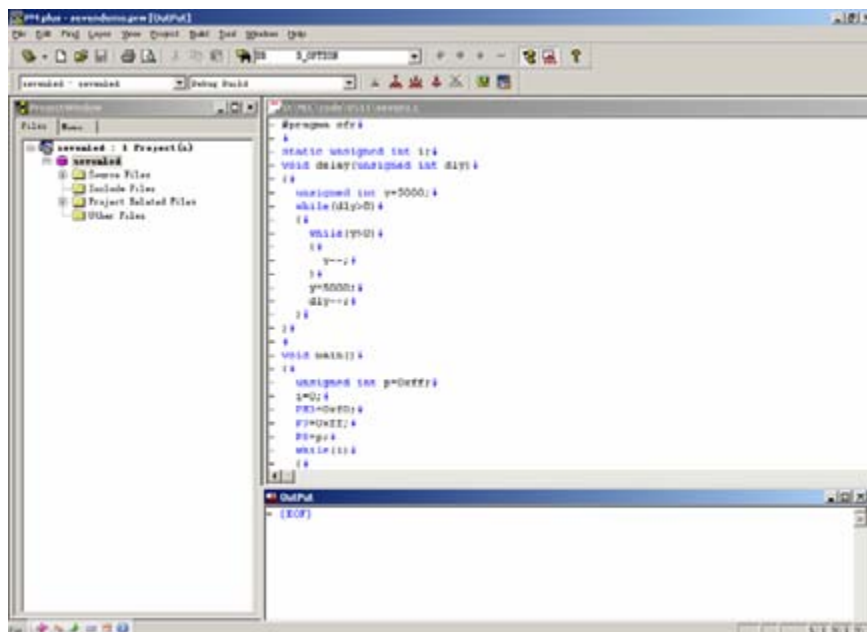


3. 选择 sevendemo.prw 后点击“打开”

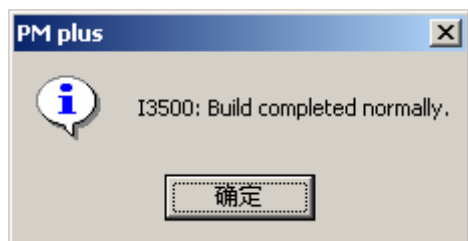


4. Project Window 中显示一系列相关的文件和函数。点击一个文件或函数之后，这个文件或函数的描述就显示在 source window 中。要编辑程序，可直接在 source window 中编辑。


5. 要编译和汇编，点击  (build icon) 。

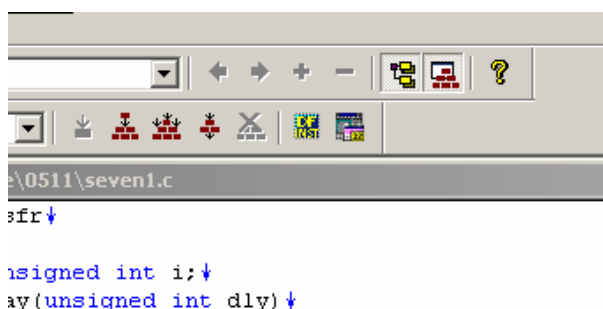


6. 编译和汇编完成之后，显示“Build completed normally.”。点击 OK 按钮。如果有错误发生，会显示“A builderror was found.”。在这种情况下，纠正错误的信息在 Output 窗口中显示。



3.2 启动 SM+ 和仿真执行

1. 点击 PM+ 中的  Simulator Start Icon 启动 SM+



2. 启动 SM+ 后，显示如图所示的窗口。点击  不同的执行按钮可

WinBox for 32-bit Windows

File Edit View Options Data Format Breakpoints Search Windows Help

Source (server.c)

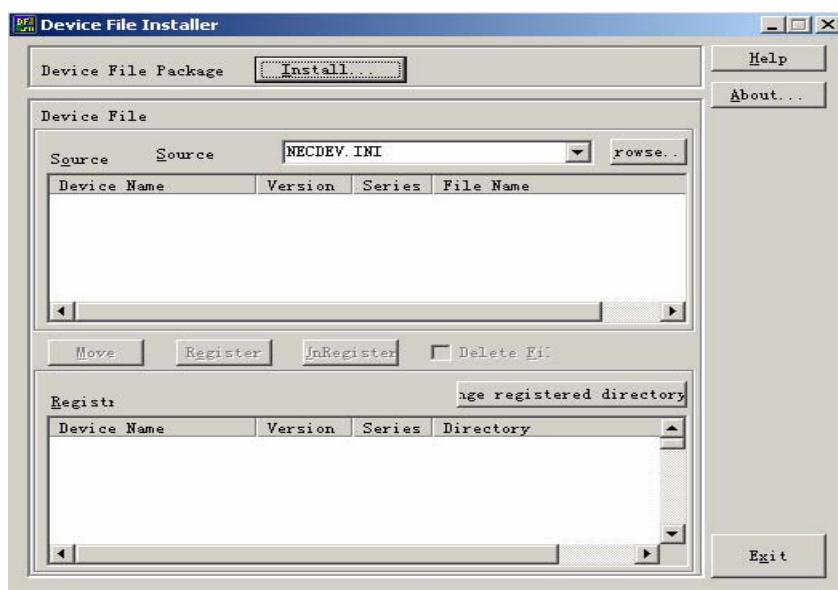
```

1  char buf[1024];
2  while(1)
3  {
4      r = read(sock, buf, 1024);
5      if(r < 0)
6          continue;
7      if(r == 0)
8          break;
9      write(sock, buf, r);
10 }
11
12 int main()
13 {
14     int sock;
15     struct sockaddr_in serv_addr;
16     sock = socket(AF_INET, SOCK_STREAM, 0);
17     if(sock < 0)
18     {
19         perror("socket");
20         return 1;
21     }
22     serv_addr.sin_family = AF_INET;
23     serv_addr.sin_port = htons(8080);
24     inet_aton("127.0.0.1", &serv_addr.sin_addr);
25     if(bind(sock, (struct sockaddr*)&serv_addr, sizeof(serv_addr)) < 0)
26     {
27         perror("bind");
28         return 1;
29     }
30     listen(sock, 5);
31     while(1)
32     {
33         int new_sock;
34         struct sockaddr_in client_addr;
35         new_sock = accept(sock, (struct sockaddr*)&client_addr, &client_addr.sin_len);
36         if(new_sock < 0)
37             continue;
38         fork();
39         if(fork() < 0)
40             continue;
41         if(fork() < 0)
42             continue;
43         if(fork() < 0)
44             continue;
45         if(fork() < 0)
46             continue;
47         if(fork() < 0)
48             continue;
49         if(fork() < 0)
50             continue;
51         if(fork() < 0)
52             continue;
53         if(fork() < 0)
54             continue;
55         if(fork() < 0)
56             continue;
57         if(fork() < 0)
58             continue;
59         if(fork() < 0)
60             continue;
61         if(fork() < 0)
62             continue;
63         if(fork() < 0)
64             continue;
65         if(fork() < 0)
66             continue;
67         if(fork() < 0)
68             continue;
69         if(fork() < 0)
70             continue;
71         if(fork() < 0)
72             continue;
73         if(fork() < 0)
74             continue;
75         if(fork() < 0)
76             continue;
77         if(fork() < 0)
78             continue;
79         if(fork() < 0)
80             continue;
81         if(fork() < 0)
82             continue;
83         if(fork() < 0)
84             continue;
85         if(fork() < 0)
86             continue;
87         if(fork() < 0)
88             continue;
89         if(fork() < 0)
90             continue;
91         if(fork() < 0)
92             continue;
93         if(fork() < 0)
94             continue;
95         if(fork() < 0)
96             continue;
97         if(fork() < 0)
98             continue;
99         if(fork() < 0)
100             continue;
101         if(fork() < 0)
102             continue;
103         if(fork() < 0)
104             continue;
105         if(fork() < 0)
106             continue;
107         if(fork() < 0)
108             continue;
109         if(fork() < 0)
110             continue;
111         if(fork() < 0)
112             continue;
113         if(fork() < 0)
114             continue;
115         if(fork() < 0)
116             continue;
117         if(fork() < 0)
118             continue;
119         if(fork() < 0)
120             continue;
121         if(fork() < 0)
122             continue;
123         if(fork() < 0)
124             continue;
125         if(fork() < 0)
126             continue;
127         if(fork() < 0)
128             continue;
129         if(fork() < 0)
130             continue;
131         if(fork() < 0)
132             continue;
133         if(fork() < 0)
134             continue;
135         if(fork() < 0)
136             continue;
137         if(fork() < 0)
138             continue;
139         if(fork() < 0)
140             continue;
141         if(fork() < 0)
142             continue;
143         if(fork() < 0)
144             continue;
145         if(fork() < 0)
146             continue;
147         if(fork() < 0)
148             continue;
149         if(fork() < 0)
150             continue;
151         if(fork() < 0)
152             continue;
153         if(fork() < 0)
154             continue;
155         if(fork() < 0)
156             continue;
157         if(fork() < 0)
158             continue;
159         if(fork() < 0)
160             continue;
161         if(fork() < 0)
162             continue;
163         if(fork() < 0)
164             continue;
165         if(fork() < 0)
166             continue;
167         if(fork() < 0)
168             continue;
169         if(fork() < 0)
170             continue;
171         if(fork() < 0)
172             continue;
173         if(fork() < 0)
174             continue;
175         if(fork() < 0)
176             continue;
177         if(fork() < 0)
178             continue;
179         if(fork() < 0)
180             continue;
181         if(fork() < 0)
182             continue;
183         if(fork() < 0)
184             continue;
185         if(fork() < 0)
186             continue;
187         if(fork() < 0)
188             continue;
189         if(fork() < 0)
190             continue;
191         if(fork() < 0)
192             continue;
193         if(fork() < 0)
194             continue;
195         if(fork() < 0)
196             continue;
197         if(fork() < 0)
198             continue;
199         if(fork() < 0)
200             continue;
201         if(fork() < 0)
202             continue;
203         if(fork() < 0)
204             continue;
205         if(fork() < 0)
206             continue;
207         if(fork() < 0)
208             continue;
209         if(fork() < 0)
210             continue;
211         if(fork() < 0)
212             continue;
213         if(fork() < 0)
214             continue;
215         if(fork() < 0)
216             continue;
217         if(fork() < 0)
218             continue;
219         if(fork() < 0)
220             continue;
221         if(fork() < 0)
222             continue;
223         if(fork() < 0)
224             continue;
225         if(fork() < 0)
226             continue;
227         if(fork() < 0)
228             continue;
229         if(fork() < 0)
230             continue;
231         if(fork() < 0)
232             continue;
233         if(fork() < 0)
234             continue;
235         if(fork() < 0)
236             continue;
237         if(fork() < 0)
238             continue;
239         if(fork() < 0)
240             continue;
241         if(fork() < 0)
242             continue;
243         if(fork() < 0)
244             continue;
245         if(fork() < 0)
246             continue;
247         if(fork() < 0)
248             continue;
249         if(fork() < 0)
250             continue;
251         if(fork() < 0)
252             continue;
253         if(fork() < 0)
254             continue;
255         if(fork() < 0)
256             continue;
257         if(fork() < 0)
258             continue;
259         if(fork() < 0)
260             continue;
261         if(fork() < 0)
262             continue;
263         if(fork() < 0)
264             continue;
265         if(fork() < 0)
266             continue;
267         if(fork() < 0)
268             continue;
269         if(fork() < 0)
270             continue;
271         if(fork() < 0)
272             continue;
273         if(fork() < 0)
274             continue;
275         if(fork() < 0)
276             continue;
277         if(fork() < 0)
278             continue;
279         if(fork() < 0)
280             continue;
281         if(fork() < 0)
282             continue;
283         if(fork() < 0)
284             continue;
285         if(fork() < 0)
286             continue;
287         if(fork() < 0)
288             continue;
289         if(fork() < 0)
290             continue;
291         if(fork() < 0)
292             continue;
293         if(fork() < 0)
294             continue;
295         if(fork() < 0)
296             continue;
297         if(fork() < 0)
298             continue;
299         if(fork() < 0)
300             continue;
301         if(fork() < 0)
302             continue;
303         if(fork() < 0)
304             continue;
305         if(fork() < 0)
306             continue;
307         if(fork() < 0)
308             continue;
309         if(fork() < 0)
310             continue;
311         if(fork() < 0)
312             continue;
313         if(fork() < 0)
314             continue;
315         if(fork() < 0)
316             continue;
317         if(fork() < 0)
318             continue;
319         if(fork() < 0)
320             continue;
321         if(fork() < 0)
322             continue;
323         if(fork() < 0)
324             continue;
325         if(fork() < 0)
326             continue;
327         if(fork() < 0)
328             continue;
329         if(fork() < 0)
330             continue;
331         if(fork() < 0)
332             continue;
333         if(fork() < 0)
334             continue;
335         if(fork() < 0)
336             continue;
337         if(fork() < 0)
338             continue;
339         if(fork() < 0)
340             continue;
341         if(fork() < 0)
342             continue;
343         if(fork() < 0)
344             continue;
345         if(fork() < 0)
346             continue;
347         if(fork() < 0)
348             continue;
349         if(fork() < 0)
350             continue;
351         if(fork() < 0)
352             continue;
353         if(fork() < 0)
354             continue;
355         if(fork() < 0)
356             continue;
357         if(fork() < 0)
358             continue;
359         if(fork() < 0)
360             continue;
361         if(fork() < 0)
362             continue;
363         if(fork() < 0)
364             continue;
365        
```

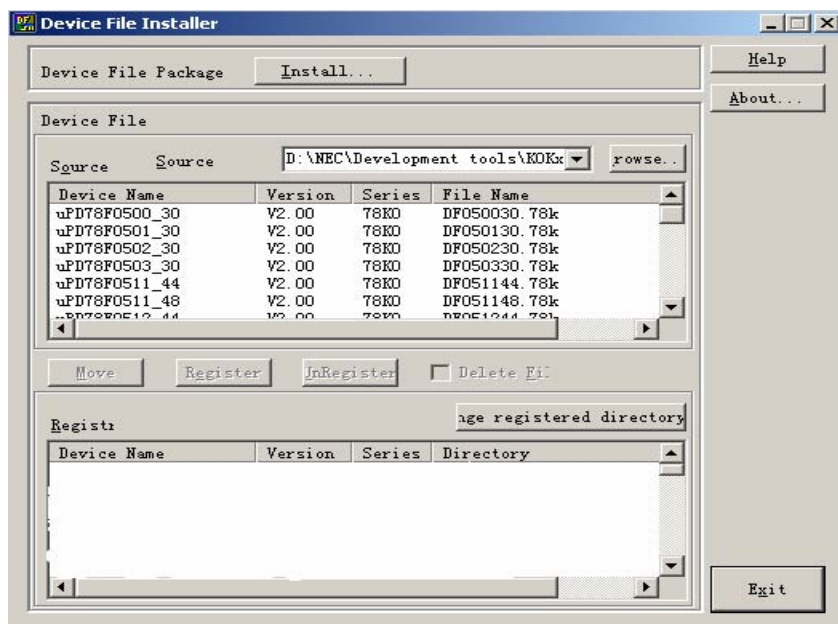
4. 要暂停程序的执行, 点击  按钮, 若要按步执行, 点击  按钮。

4.实验板的使用方法

〈1〉 单击, 开始-程序-NEC Tools32-Device File Installer

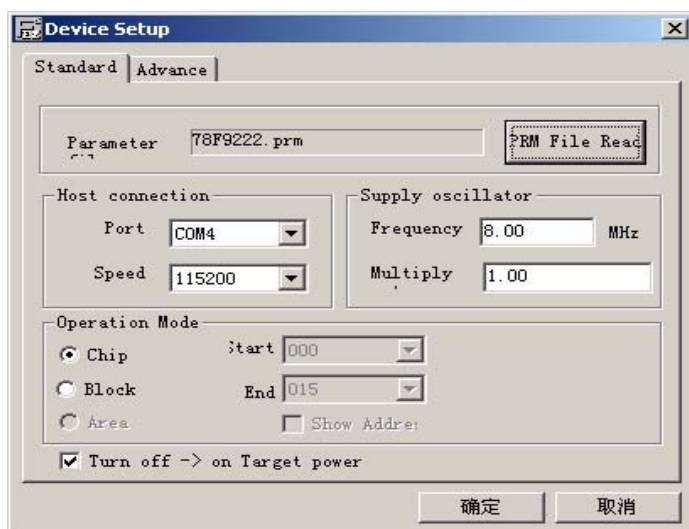


4



〈3〉 选择 uPD78F9222，单击 **Register**，参数文件安装成功。

- 2 打开 PM Plus，载入工程，编译。（详细操作请查看前面章节）
- 3 把实验板上的跳线 JP1—JP3 跳至 TO RUN 处，即 2, 3 短路（实验板上有详细标识）
- 4 连接实验板的 USB
- 5 打开 WriteEZ2 软件，载入烧写参数文件，选择通信端口，具体设置如下图

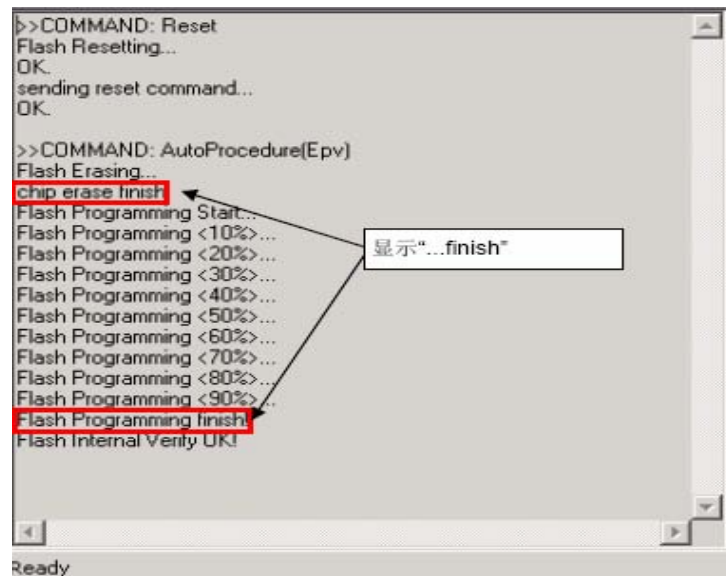


- 6  打开需要烧写的 HEX 文件（此文件在你所建的工程文件夹下）

- 7 单击 ，出现该提示框



- 8 把实验板上的跳线 JP1—JP3 跳至 T0 PC 处，即 1,2 短路（实验板上有详细标识）后，
按确定
实验板就会进行烧写，显示工作进度



- 9 完成后，断掉 USB，把跳线短接至 2—3（即 T0 RUN），实验板开始工作。