```asm
 1
 2 parse_execute_command: // La diferencia con el anterior es que a esta rutina la llamo desde el modo manual
 3
 4          cpi IN_COMMAND, 's'              // STAND BY
 5          breq stand_by_jump
 6
 7          cpi IN_COMMAND, 'a'              // MODO AUTOMATICO
 8          breq receive_auto_mode
 9
10          cpi IN_COMMAND, 'b'              // BACKSPIN
11          breq set_backspin_call
12
13          cpi IN_COMMAND, 't'              // TOPSPIN
14          breq set_topspin_call
15
16          cpi IN_COMMAND, 'v'              // SET FIRE SPEED
17          breq receive_fire_speed
18
19          cpi IN_COMMAND, 'n'              // SER FIRE ANGLE
20          breq receive_fire_angle
21
22          cpi IN_COMMAND, 'd'
23          breq shoot_jump
24
25          cpi IN_COMMAND, 'm'
26          breq start_mixer_call
27
28          jmp await_command
29
30 receive_fire_speed:
31          call usart_recieve_command
32          mov PARAMETER, IN_COMMAND
33          call set_ball_speed
34      ret
35
36 receive_fire_angle:
37          call usart_recieve_command
38          mov PARAMETER, IN_COMMAND
39          call set_fire_angle
40      ret
41
42 receive_auto_mode:
43          call usart_recieve_command
44          mov PARAMETER, IN_COMMAND
45
46          // Limpio la interrupcion
47          ldi r16, (1 << RXC0)
48          lds r17, UCSR0A
49
50          or r16, r17
51          sts UCSR0A, r16
52
53          mov PARAMETER, IN_COMMAND
54      jmp automatic_mode
55
56
57 set_topspin_call:
58          call set_topspin
59          ret
60
61 stand_by_jump:
62          jmp stand_by
63
64
65 set_backspin_call:
66          call set_backspin
67          ret
68
69 shoot_jump:
```

```
70          call shoot
71          ret
72
73 start_mixer_call:
74          call start_mixer
75          ret
```