macchina a stati finiti della SCO del tastierino:

global variables{

}

events{

status\_on;

status\_off;

int\_req\_0;

return\_sca;

}

commands{

turn\_on;

set\_int\_req\_1;

set\_int\_req\_0;

set\_status\_1;

set\_status\_0;

print\_schermo\_message;

}

#accensione: microprogramma della sco che parte all'avvio della periferica

# e non può essere raggiunto in un secondo momento

# viene preparata la sca e settati i flip-flop

state off{

actions{set\_status\_0};

=>prepeare\_turned\_on;

}

state prepeare\_turned\_on{

actions{turn\_on, set\_int\_req\_0};

=>turned on;

}

state turned\_on{

actions{set\_status\_1}; #nel caso in cui la cpu sia in busy waiting in attesa che la

#periferica si accenda, con questo comando comunichiamo alla

#cpu che la periferica è pronta

=>idle;

}

state idle{

actions{set\_status\_0};

status\_on=>wait\_return;

}

#insert

state wait\_return{

actions{set\_status\_0, start\_sca};

return\_sca=>interrupt;

}

#interrupt

state interrupt{

actions{set\_status\_1, set\_int\_req\_1, set\_iterator\_0};

status\_off=>wait;

}

state wait{

actions{set\_int\_req\_0};

status\_on=>idle;

}

#gestione errore

state error{

actions{print\_schermo\_message};

}