A simple interlocked controller for research vacuum systems

2001 Meas. Sci. Technol. 12 N43

(http://iopscience.iop.org/0957-0233/12/12/402)

Meas. Sci. Technol. **12** (2001) N43–N46 www.iop.org/Journals/mt PII: S0957-0233(01)27418-9

DESIGN NOTE

A simple interlocked controller for research vacuum systems

Tim Freegarde[1], **Jim Jessup and Gus Hancock**

Physical and Theoretical Chemistry Laboratory, South Parks Road, Oxford OX1 3QZ, UK

E-mail: freegard@science.unitn.it

Received 23 July 2001, in final form 25 September 2001, accepted for publication 19 October 2001

Abstract
We describe a simple, low-cost circuit that we use to control and supervise the operation of a radio-frequency plasma chamber. The unit has been designed for flexibility and ease of construction, all control logic being programmed into read-only memories. Feedback allows the unit to operate as a state machine, offering a limited degree of control complexity without the need for more than a simple EPROM programmer.

Keywords: controller, interlock, vacuum apparatus

1. Introduction

The start-up procedure for complex vacuum systems can be time-consuming if carried out manually, yet commercial control units tend to be expensive; in a research environment, their use is rare. The controller described here is routinely used to run a radio-frequency plasma system, comprising the chamber with pressure gauges and temperature interlocks, rotary and turbo pumps, mass-flow and pressure controllers, an isolating gate valve, and the RF power supply. In the event of a failure whilst running, the controller falls back to one of several safe conditions, depending upon the severity of the problem. All interlock inputs and control outputs may be manually overridden. Inputs may be control voltages or contact closures, of either polarity; outputs are contact closures that may also be wired as switched mains supplies.

This controller provides an compromise between the simplicity of hard-wired electro-mechanical controllers and the flexibility of expensive microprocessor-based designs. The control operation is stored in easily programmed UV erasable read-only memory (EPROM).

2. Operation of the plasma system

To provide a clear illustration of the operation of this unit, we describe the specific system, shown in figure 1, for which it was designed. The inductively coupled plasma reactor comprises

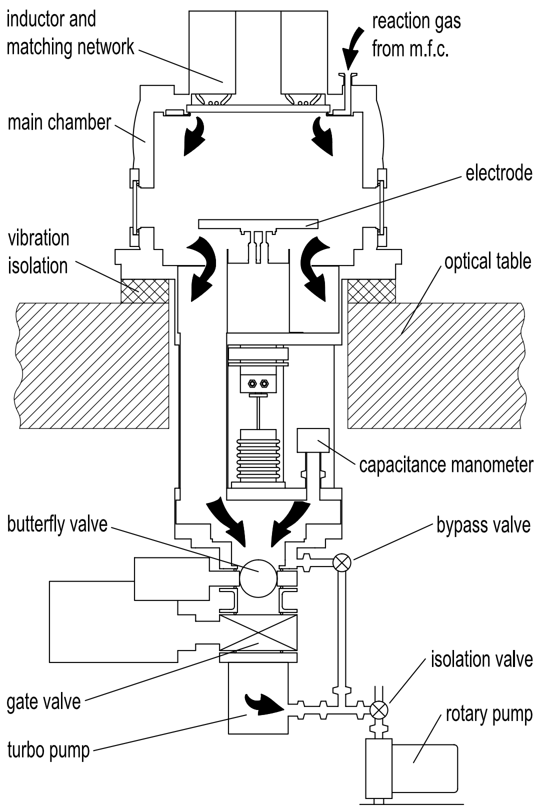

Figure 1. The inductively coupled radio-frequency plasma chamber and associated apparatus.

[1] Present address: Dipartimento di Fisica, Università di Trento, Via Sommarive 14, 38050 Povo (TN), Italy

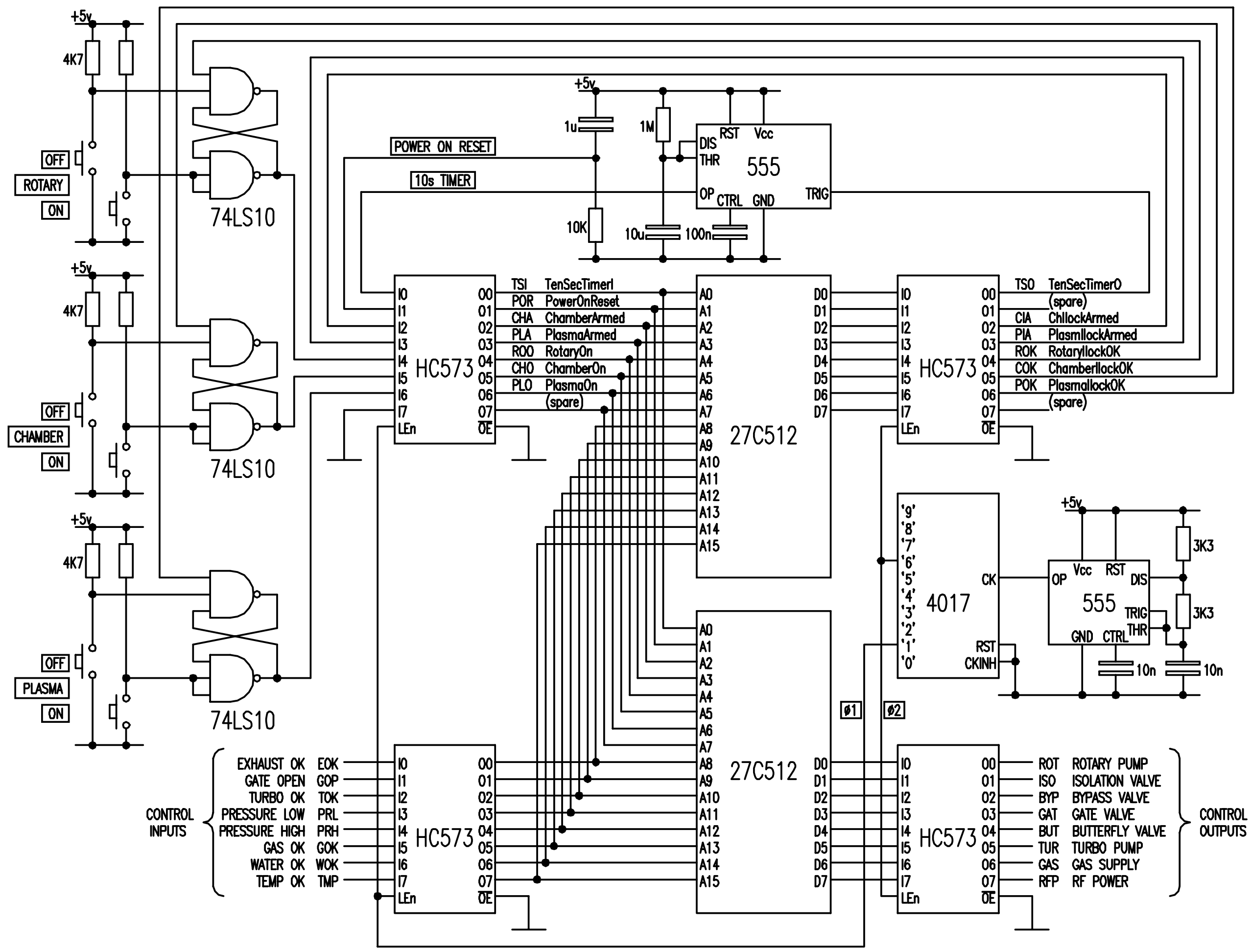

Figure 2. Schematic circuit diagram of the main controller logic. The circuit is based around a pair of EPROMs, with input and output latches driven in antiphase, which acts as a look-up table to convert inputs into output signals. Three inputs from user-set bistables and a fourth from a 10 second timer may be reset by the controller. Feedback from the outputs to the inputs allows internal states that correspond to data bits, allowing simple non-unitary processing.

a main chamber that is evacuated by a turbo pump backed by a rotary pump. The chamber can be isolated by a gate valve, and the exhaust is modulated by a butterfly valve which can be held fully open for initial pumping but which, in normal operation, is linked via a pressure controller to a capacitance diaphragm gauge on the chamber. Gas is supplied to the chamber via a mass-flow controller, and radio-frequency power from a switchable power supply is coupled into a planar inductor in the chamber lid. Electrically operated valves isolate and vent the rotary pump and bypass the turbo pump for initial rough evacuation. Interlocks monitor coolant flow, chamber temperature and gas supply, and the turbo pump and vacuum gauge controllers provide monitors of gauge pressures and pump status.

For routine operation we have three operation modes, which correspond to

- maintaining a rough vacuum in the exhaust line through rotary pump operation alone
- normal pumping of the plasma chamber, using the turbo and rotary pumps
- complete operation of the plasma chamber with a pressure- and mass-flow-controlled gas supply and radio-frequency excitation of the plasma.

In the event of a failure, such as an error condition from the turbo pump or the tripping of a temperature sensor, the system drops to a lower mode of operation that depends upon the severity of the problem.

3. Controller logic

The main part of the controller is shown in figure 2, and is based around a pair of 512k read-only memories (27C512) whose inputs and outputs are latched in antiphase at a clock frequency of a few kilohertz. For much of the circuit operation, the EPROMs function as a look-up table, producing a set of outputs that depends upon the input conditions from the various system interlocks and from three control inputs, set or reset by the user and resettable by the system, which define the desired mode of operation. A short pulse at power-up is applied to an additional input to define the starting state of the system, and we also include a timer that allows a delay between starting the rotary pump and connecting it to the vacuum system. Three further inputs are connected to outputs of the look-up table, allowing the controller to operate as a state machine [1]: depending upon the programming, the system

Table 1. Partial truth table showing state machine operation. The outputs `ChamberInterlockArmed` and `ChamberInterlockOK` are returned, via the latches, to the inputs `ChamberArmed` and `ChamberOn` respectively, the latter passing via the user-switchable bistable circuit.

State	Rotary On	System OK	Chamber On	Chamber Armed	ChamberInterlock OK	ChamberInterlock Armed	Stable	Next	Description
0	0	0	0	0	0	0	•	0	system off
1	0	0	0	1	0	0		0	
2	0	0	1	0	0	0		0	
3	0	0	1	1	0	1		1	
4	0	1	0	0	0	0	•	4	evacuated but off
5	0	1	0	1	0	0		4	
6	0	1	1	0	0	1		5	
7	0	1	1	1	0	1		5	
8	1	0	0	0	0	0	•	8	rotary on
9	1	0	0	1	0	0		8	chamber turning off
10	1	0	1	0	1	0	•	10	initial pump down
11	1	0	1	1	0	1		9	interlock failure
12	1	1	0	0	0	0	•	12	system ok but off
13	1	1	0	1	0	0		12	chamber switched off
14	1	1	1	0	1	1		15	pump down complete
15	1	1	1	1	1	1	•	15	all well

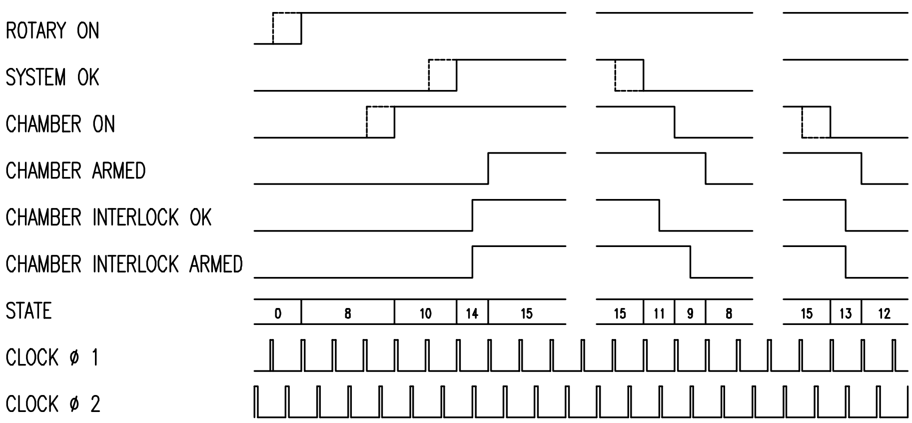

Figure 3. Timing diagram showing state machine operation via feedback. Only once normal running has been achieved may the monitor interlocks interrupt operation. All signals are shown at the latch outputs (as labelled in figure 2); `ROTARY ON`, `CHAMBER ON` and `CHAMBER ARMED` are inputs to the EPROM, while `CHAMBER INTERLOCK OK` and `CHAMBER INTERLOCK ARMED` are outputs. `SYSTEM OK` represents a combination of inputs, as explained in the text.

may thereby have extra, internal states which may be regarded as data bits and which therefore allow a degree of non-unitary processing. Common applications of such techniques tend to require specific development and programming units; in contrast, the implementation described here requires nothing more complicated than an EPROM programmer.

The state-machine operation is illustrated in table 1, which shows a partial truth table for the memory programming of our system. The table shows the dependence of two outputs `ChamberInterlockOK` (`COK`) and `ChamberInterlockArmed` (`CIA`) upon four inputs: `RotaryOn` (`ROO`), `ChamberOn` (`CHO`), `ChamberArmed` (`CHA`) and `SystemOK`, where

$$\mathtt{SystemOK} = \mathtt{PressureLow} \cdot \mathtt{ExhaustOK} \cdot \mathtt{WaterOK} \quad (1)$$

and it is assumed throughout that the `PowerOnReset` (`POR`)=0. The truth table corresponds to the logical definitions

$$\mathtt{COK} = \mathtt{ROO} \cdot \mathtt{CHO} \cdot \left(\overline{\mathtt{CHA}} \times \mathtt{SystemOK}\right) \cdot \overline{\mathtt{POR}}$$
$$\mathtt{CIA} = \mathtt{CHO} \cdot (\mathtt{CHA} \times \mathtt{SystemOK}) .$$

`ChamberInterlockArmed` (`CIA`) is fed back directly to the `ChamberArmed` (`CHA`) input, and `ChamberInterlockOK` (`COK`) returns to `ChamberOn` (`CHO`) via the bistable, thus determining to some extent the state after the following clock cycle. There are six stable states, indicated in table 1; all other states cascade down to one of these within a few clock cycles. Figure 3 shows a timing and state diagram for normal start-up and shut-down as well as the response to failure of one of the interlocks comprising `SystemOK`.

In this example, the internal state `CIA` allows the controller to ignore the chamber and exhaust pressures, which may be high while the system is starting up, until normal running has been achieved. From its initial state 0, the system is turned on by pushing the *Rotary On* switch, the subsequent configuration (state 8) corresponding to the first of our three modes of operation. Pressing *Chamber On* then causes a transition to state 10 but, until the chamber has been evacuated, `SystemOK` will remain false and the chamber interlock will not be satisfied. The controller ignores this until `SystemOK` becomes active, when the interlock is armed and the system switches to state 15, the second mode of operation, starting the turbo pump. Should the conditions represented by `SystemOK` now cease to be met, the armed system will respond by returning to

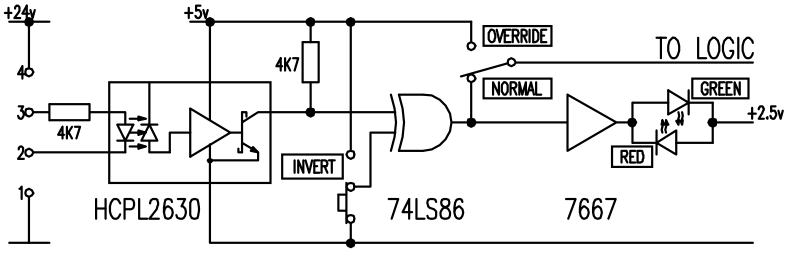

Figure 4. Generic input stage, allowing both logic level and contact closure signals of either polarity.

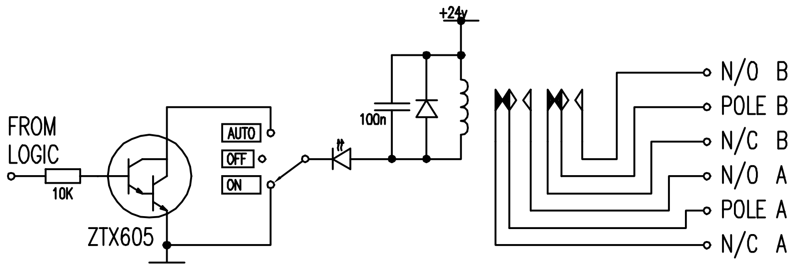

Figure 5. The output drives a DPDT relay, and may be manually overridden to be set either on or off. Operation is indicated by an LED.

state 8, the first mode of operation, via states 11 and 9. Even should `SystemOK` then return to being true, the second mode of operation will not be entered until the *Chamber On* switch is pressed again. Our third mode of operation is controlled similarly via `PlasmaInterlockArmed` etc.

The complete logical definition of the controller[2] in our specific case is given below, using the abbreviations shown in figure 2:

$$
\begin{aligned}
&\mathtt{ROT} = \mathtt{ROO}\\
&\mathtt{TSO} = \overline{\mathtt{ROO}\cdot\overline{\mathtt{POR}}}\\
&\mathtt{ISO} = \mathtt{ROO}\cdot\overline{\mathtt{TSI}}\\
&\mathtt{BYP} = \mathtt{CHO}\cdot\overline{\mathtt{TSI}}\cdot\mathtt{PRH}\\
&\mathtt{GAT} = \mathtt{CHO}\cdot\overline{\mathtt{TSI}}\cdot\mathtt{PRL}\\
&\mathtt{TUR} = \mathtt{CHO}\cdot\mathtt{PRL}\cdot\mathtt{WOK}\cdot\mathtt{EOK}\cdot\mathtt{GOP}\\
&\mathtt{BUT} = \mathtt{CHO}\cdot\mathtt{TOK}\\
&\mathtt{GAS} = \mathtt{PLO}\cdot\mathtt{TOK}\cdot\mathtt{GOP}\cdot\mathtt{GOK}\\
&\mathtt{RFP} = \mathtt{PLO}\cdot\mathtt{TOK}\cdot\mathtt{GOP}\cdot\mathtt{GOK}\cdot\mathtt{TMP}\\
&\mathtt{ROK} = \overline{\mathtt{POR}}\\
&\mathtt{COK} = \mathtt{ROO}\cdot\mathtt{CHO}\cdot\big(\overline{\mathtt{CHA}}\times(\mathtt{PRL}\cdot\mathtt{WOK}\cdot\mathtt{EOK})\big)\\
&\mathtt{CIA} = \mathtt{CHO}\cdot(\mathtt{CHA}\times(\mathtt{PRL}\cdot\mathtt{WOK}\cdot\mathtt{EOK}))\\
&\mathtt{POK} = \mathtt{CHO}\cdot\mathtt{PLO}\cdot\big(\overline{\mathtt{PLA}}\times(\mathtt{PRL}\cdot\mathtt{TOK})\big)\\
&\mathtt{PIA} = \mathtt{PLO}\cdot(\mathtt{PLA}\times(\mathtt{PRL}\cdot\mathtt{TOK}))\,.
\end{aligned}
$$

4. Input and output

Input signals for the controller are derived from a variety of sources: the turbo pump controller produces logic-level output voltages to indicate the state of operation, while pressure gauges, flow switches and so on instead offer contact closure. We therefore provide each channel with an optically isolated input circuit, shown in figure 4, that can be wired for either of these cases and either logic polarity. For low-level logic signals, a smaller input resistor may be required.

The output channels of our controller, shown in figure 5, must cope with a similar variety of output requirements, and are thus buffered by a Darlington transistor pair to drive a double pole change-over relay, which may subsequently switch mains or d.c. voltages as required. Each of the input channels may be overridden into the 'true' state, and the outputs may similarly be manually switched either on or off.

Acknowledgments

The authors wish to thank the UK Engineering & Physical Sciences Research Council for support during the course of this work.

Reference

[1] Horowitz P and Hill W 1989 *The Art of Electronics* (Cambridge: Cambridge University Press)

[2] The C program (pcontr.cpp) used to generate the memory maps from the logical definition is available as a multimedia enhancement from the online version of this journal.