UCID- 19161

MASTER

A PNEUMATIC SAMPLE-TRANSFER SYSTEM FOR USE WITH
THE LAWRENCE LIVERMORE NATIONAL LABORATORY
ROTATING TARGET NEUTRON SOURCE (RTNS-I)

Robert E. Williams
Nuclear Chemistry Division

July 1981

Lawrence
Livermore
Laboratory

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore Laboratory under Contract W-7405-Eng-48.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

FAST-NEUTRON IRRADIATIONS

A sample-transfer system is absolutely essential if one is to irradiate many samples with 14-MeV neutrons at the RTNS-I. Not only is it very inconvenient to shut down the accelerator and go into the target pit to retrieve samples, constant repetition of such a practice poses an extreme radiation hazard. In addition, the rabbit system is able to retrieve samples in just three seconds, making possible measurements of short-lived radioisotopes. For these reasons, the rabbit system was installed at the RTNS-I in the late 1960's and has twice been modified.

The system consists of the rabbit loader, an irradiation station, and the rabbit dropout. The sample loader, pictured in Fig. 1, also has a rabbit catcher and plunger to position a sample before a Ge(Li) detector. From this point, the rabbit may easily be returned to the RTNS-I for a second irradiation. Thus, samples may be cycled back and forth for irradiation and counting as often as needed. The irradiation station, Fig. 2, is colinear with the deuteron beam, and can be positioned about 0.5 mm from the rotating target. Rabbits are held in place by air pressure, and in the event of a rabbit system malfunction, the rabbit will fall away from the neutron source. Rabbits are removed from the system at the sample dropout. They travel between the sample loader and the irradiation station through 65 m of polyethylene tubing. Figure 3 is a schematic of the entire system.

Originally, small incandescent lamps and photosensors were used to detect the arrival of a rabbit, signaling to shut off the air supply. Because these controls grew very unreliable, an LSI-11 was incorporated and software developed to control the system. Infra-red photosensors were installed also,

because they are not adversely affected by dust as were the visible-light sensors, and because the infra-red sensors easily clamp around the outside of the polyethylene tube. The above modifications, completed in November 1980, have resulted in a very reliable system.

Rabbits remaining from the original system also were the source of problems. Lids, which snapped over the end of the rabbit to hold the samples in place, frequently were removed in transit, getting stuck in the tubing and sometimes spilling the samples. The design of these rabbits was complex and fabrication costs were high. A new rabbit was designed, therefore, with a snap-in cap (yet to fail in 2500 cycles) and a more simple design. Sample capsules, somewhat smaller than those in use with earlier rabbits, are about 1.4 cm in diameter and 0.1 to 0.5 cm thick. The largest capsule, with a sample volume of, 0.7 cm^3 , is adequate for most purposes. A rabbit, cap, and sample capsule are shown in Fig. 4.

Low-density polyethylene was used for the rabbits because it contains (in principle) no elements with proton numbers greater than 6, and because the low-density polyethylene is much less brittle than high-density polyethylene, which tended to break apart in the system. For analysis of short-lived radioisotopes, it is crucial that there be little oxygen in the polyethylene. Upon 14-MeV neutron irradiation, ^{16}N is produced from ^{16}O . The short half life, 7.2 s, and very high-energy gamma ray, 6.1 MeV, of ^{16}N can cause great problems in measurements of short-lived activities. Blank rabbits are not entirely clean, however, as ^{28}Al and ^{27}Mg activity, from Si and Al, are easily detected after short irradiations. Most of the contamination is on the outside of the rabbits, but there is activity from blank sample capsules indicating trace quantities of Al and Si on or in the polyethylene capsules.

Polyethylene rods were purchased from Cadillac Plastics and the rabbits were machined by Beimer's Machine Works in Pleasanton (PO # 1503101, Job Order 639233); contact Joe Bohac (Nuclear Chemistry Division) for details.

The irradiation geometry is depicted in Fig. 5. About 0.5 cm separates the sample from the actual neutron source, the deuteron beam spot on the rotating target. Although the maximum fast-neutron flux from the RTNS-I is about 2.0×10^{12} n/cm²-s, the flux decreases rapidly with the distance (1,2). It was therefore necessary to measure the flux through the sample volume. A capsule filled with Fe₂O₃ was irradiated producing ⁵⁶Mn. Using a value of 103 ± 6 mb for the ⁵⁶Fe(n, p) ⁵⁶Mn cross section at 14.9 MeV, the average flux was determined to be 2.0×10^{11} n/cm²-s at a RTNS-I source strength of 2.7×10^{12} n/s. The average neutron energy in the forward direction (along the beam line) is 14.9 MeV. Since many fast-neutron cross sections vary significantly between 13 and 15 MeV (3), care must be taken when employing nominally 14-MeV neutron cross-section data.

SYSTEM DESCRIPTION

A rabbit is placed in the loader, with the sample facing forward, and sent to the RTNS-I target by activating Valve A while the default position of Valve B is the exhaust. It is held in place during irradiation by low pressure flow through Valve E. To return the rabbit, Valve B is actuated with Valves A and E in the exhaust position. A switch (activated by Valve G) allows the removal of the rabbit from the system at the dropout or placement of the sample before a detector in the plunger, where it can be cycled through another irradiation. Valves H and J operate the dropout and the plunger, respectively.

Five sensors are used by computer programs operating the system to detect the presence of a rabbit and actuate valves accordingly. Sensors 1 through 4 are infra-red photosensors that clamp around the outside of the rabbit tube, and can detect a rabbit through the polyethylene walls. Two sensors are at the target end; a backup exists to avoid replacing a sensor in a high radiation field. Sensor 5 is a microswitch activated only when a rabbit has been placed before the detector.

An LSI-11 has been programmed in PASCAL to operate the rabbit system in either a user-controlled mode or an automated cyclic-activation mode. Documentation for the LSI-11 and its interface to the system valves is available from the Nuclear Chemistry Division Electronics Engineering Shop. The computer controls all the valves, interrogates the sensors, sets irradiation and counting times, and gates the detector when the count should begin according to whether or not the irradiation sequence was correctly executed. Any gamma-ray spectrometer may be used; the only interface between the LSI-11 and the spectrometer is the +5 volt gate.

System software is described in Appendix C. Instructions in the following section will enable a user to start the LSI-11 and operate the rabbit system without consulting this appendix. Familiarity with the RT-11 operating system is required to make any changes in the software. Digital Equipment Corp. provides many manuals available at the LLNL Small Computer User's Library. Most useful among these are the Introduction to RT-11 and the RT-11 User's Guide. The PASCAL implementation is described in the OMSI PASCAL-1 Manual, Version 1.1 for RT-11.

INSTRUCTIONS

Prior to operation of the Rabbit System, the accelerator operators should be notified. The system should be thoroughly tested before it is moved into position at the RTNS-I target. A small distance, about 0.5 mm, should separate the rabbit irradiation end from the target so vibrations will not cause the end cap to touch the target. To operate the system, apply the air pressure and electric power, and start the LSI-11.

A. Apply Air Pressure

Air pressure in the rabbit delivery line must NEVER exceed 35 psig as measured by the gauge atop the source tank under the stairs in the RTNS-I target pit. The large regulator on the supply line to the tank is used to adjust the pressure. Before each experiment, the trap at the source end of the supply line should be cleared of rust accumulated from the Building 212 supply. The operators should be notified about any operations in the target pit.

In addition to the rabbit line, three other valves are air-pressure actuated. The solenoid valve at the plunger should receive about 10 psig. Both the rabbit line switch (on the west wall) and the dropout ball-valve need about 40 psig. The regulator for the switch is located outside the trailer where there is a valve to supply all the valves in T940.

B. Electric Power

Power to T940 is turned on at the circuit breaker inside the door to the left. All of the rabbit system valves draw power from a relay box on the west

wall of T940. The computer terminal, sensor multiplexor, valve controller, floppy-disk drive, and teletype are all plugged into the LSI-11 rack.

C. LSI-11 Rabbit System Controller

After turning on the terminal and the disk drive, place a "PASCAL System Disk" in Drive 0 and a "PASCAL Work Disk" in Drive 1. Start the LSI-11 with the power button located on its back panel. The terminal should respond with "\$". To load the operating system into the LSI-11, i.e., to Bootstrap the system, type "DX" followed by return. The user should have some experience with RT-11 to better understand the system. (If the LSI-11 does not respond with a "\$" initially, type 173000G. If there is still no "\$", turn off the LSI-11, open its container and check to see that all the boards are firmly connected to the processor back plane and that all cables to peripherals are connected. If it fails again seek professional help.)

D. Software

Each work disk should contain the Rabbit System software; for a list type DIR. Programs that run the system are in the following files:

1. BYHAND.SAV - All actions are user-initiated. Sensors are ignored.
BYHAND is used to check the air pressure needed to send rabbits, and may be useful in the event a rabbit is stuck.
2. MANUAL.SAV - The sensors are used to shut off the air when a rabbit reaches its destination. Send and return commands are required from the user. MANUAL provides a total system check-out and is the best program for long irradiations.

3. CYCLE.SAV - The rabbit is automatically cycled through a preset number of irradiation/counting sequences. A file, FIL.TXT, is created to store irradiation and counting times. A failure of any kind in a cycle will interrupt the series and ask the user whether or not to continue.
4. CYCLE6.SAV - A count-start pulse replaced the gate in CYCLE for use with the NOVA analyzer. Otherwise, CYCLE6 is the same as CYCLE.
5. CYCLES.SAV - No file is created by CYCLES, which enables restarting another series of cycles without ending the program.
6. ACYCLE.SAV - As in CYCLE, irradiation and counting times are set, but the user initiates each cycle or command. ACYCLE also creates FIL.TXT.
7. ZCYCLE.SAV - ZCYCLE is initialized to give a start pulse also, but is otherwise identical to ACYCLE.

To run one of the above programs, *****.SAV, type RUN ***** . All the programs are user-interactive; necessary actions are prompted by queries. At several points in each program, the user may abort the procedure. In the event of a disaster, type "CTRL-C" twice to terminate any program. Read the section about On-Line Debugging to be able to shut off any valves left activated following CTRL-C. Or, of course, turn off all the valves at the relay box power switch.

(Several other files are related to the rabbit system:

1. *****.PAS - Over twenty PASCAL source listings of the main programs and external procedures needed to change the programs.

2. *****.OBJ -

Compiled versions of the above ready to
be linked together if there are NO
changes in the global variables.

None of the above is required to execute the four rabbit controller programs
so they need not appear on the user's PASCAL Work Disk.)

E. Sending Rabbits

Cycle the system without a rabbit first, testing all the components
thoroughly. Snap a cap into a rabbit and place it in the sample loader. Be
certain the cover of the loader is secure before sending a rabbit. Run the
program of your choice; all programs prompt the user for instructions.

CONCLUSION

Incorporation of an LSI-11 and infra-red photo sensors have greatly
increased the reliability of the RTNS-1 rabbit system. The instructions
describe the normal operations of the system; consult the appendix for trouble
shooting hints. Before attempting any irradiations, consult the Operational
Safety Procedure 212.11 for T940 and the Hazards Control technician for
Building 212.

ACKNOWLEDGEMENTS

The following people have contributed much to improving the rabbit system: Howard Spracklen, Dave Tinoko, and Ken DeWitt, for their help with the LSJ-11 and controls; Dennis Chakedis and Gary Power for making modification to the rabbit lines; Richard Jesse for help with the software needed to utilize the sensor multiplexor; Dick Peterson for help in rabbit design. The author appreciates their help.

The author particularly wishes to thank Dick Meyer and Gene Henry for their support and assistance in this effort.

Energy by LLNL under contract No. W-7405-ENG-48.

REFERENCES

1. Richard A. Van Konynenburg, "Neutron Dosimetry (14 MeV) for Foil Samples Irradiated with the Lawrence Livermore National Laboratory Rotating Target Neutron Source", UCRL-51393 Rev. 1 (1974).
2. E. Goldberg, R. Griffith, and C. Logan, "Peak Flux Measurements of the LLNL High-Intensity 14-MeV Neutron Source", UCRL-51317 (1972).
3. M. Bormann, H. Neuert, and W. Scobel, "Tables and Graphs of Cross-Sections for (n,p) , (n,α) , and $(n,2n)$ Reactions in the Neutron Energy Region 1-37 MeV", Handbook on Nuclear Activation Cross-Sections, IAEA Technical Reports Series No. 156, Vienna, 1974.
4. Peter Grogono, Programming in PASCAL (Revised Edition), Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1980.

APPENDIX A: ON-LINE DEBUGGING TECHNIQUE (ODT)

ODT is a powerful tool allowing a user to step through the execution of a program, monitoring machine language by displaying memory contents. Consult the Introduction to RT-11 for a complete description of ODT. For the purpose of manipulating the rabbit system by changing the contents of a memory location, one needs to know only a few commands.

At any time, the contents of a memory location may be displayed. Simply type "BREAK" and the LSI-11 interrupts whatever it is doing and responds with "@". The user responds with the location to be displayed followed by "/", which prints the contents of the location and opens it so the user may enter new contents. Whether or not new contents are entered, "RETURN" closes the location and the LSI-11 prompts again with "@". A sequence of locations can be examined using "LINE FEED", which closes a location and opens the next.

The major use of ODT with the Rabbit System will be manual control of a valve. All of the commands for actuating or turning off valves are entered into the location 166660. Table 1 lists each valve, its function, and the value of location 166660 needed to turn the valve on or off. To accomplish this type "BREAK", then 166660/, the command from the table, and "RETURN". To return from ODT type "P" and "RETURN" (if the LSI-11 responds to this with "file?", type "CTRL-C").

TABLE 1. VALVE CONTROL COMMANDS

<u>Valve</u>	<u>Controller No.</u>	<u>Function</u>	<u>ON/OFF</u>	<u>Command</u>
A, D	0	Send rabbit	ON OFF	0 100000
B	1	Return rabbit	ON OFF	10 100010
E, F	2	Irradiation	ON OFF	20 100020
H	4	Dropout	OPEN CLOSE	40 100040
J	5	Plunger	DOWN UP	50 100050
G	6	Rabbit-line switch	DROPOUT DETECTOR	60 100060
	7	5-v detector gate	OPEN CLOSED	70 100070

APPENDIX B: TROUBLE-SHOOTING

Table 2 is a summary of several problems likely to occur upon extensive use of the rabbit system. Possible causes are listed for a number of symptoms and remedies are suggested. Read the section about sensitivity adjustments to the sensors before making such adjustments.

TABLE 2. TROUBLE-SHOOTING

TROUBLE	CAUSE	REMEDY
Sensor fails to see rabbit	Rabbit getting stuck in tube	1) Make sure that no sharp turns have been introduced by moving equipment.
		2) Check air pressure to the switch and see if it is operating properly.
		3) Check for build up of crud inside tube.
	Sensor Malfunctions	
	1) Sensitivity needs adjustment	1) Each sensor card should have its red light on with no rabbit present and off only when a rabbit breaks the beam. See the section on adjusting the sensitivity and attempt to correct the problem before trying the following remedies.
	2) Defective sensor	1) Test sensor in another circuit board. If it fails replace the sensor with another in the original board and re-adjust the sensitivity.
	3) Defective circuit boards	1) Replace green circuit board and adjust its sensitivity. 2) Return first green board changing the larger L-shaped board the sensor is connected to. Leave no blanks between the cables at either end of the series of boards!

TABLE 2. TROUBLE-SHOOTING (Continued)

TROUBLE	CAUSE	REMEDY
Sensor sees rabbit when there is none	Sensor is too sensitive	Adjust sensitivity.
	Sensor incorrectly mounted on tube	Check for obstructions, alignment.
Plunger fails to catch rabbit	Bad timing	1) Experiment with different settings of the plunger delay time set in the programs and
	Bounces out of plunger	2) Adjust the flow through the two exhaust lines at the plunger so that more flow goes through the rabbit line.
LSI-11 "freezes"	Only God knows	1) Try "CTRL-C" a few times. 2) "Break" and Type P then "Return". 3) "Break" and re-bootstrap with 173000G which should return a \$. 4) Completely shut down computer, terminal and disk drive and start again.

Adjusting Sensor Sensitivity

Each green circuit board in the sensor multiplexor contains a small variable resistance used for sensitivity adjustments. Looking at the board from the back of the multiplexor box, a CLOCKWISE turn of the screw DECREASES the sensitivity (do this if the red light is always off). The sensitivity adjustment is itself very sensitive; make small changes (less than a quarter turn) and turn off and on again the multiplexor power. If the sensor light is on but the sensor fails to see a rabbit, increase the sensitivity slightly and try again several times before a further adjustment.

If many adjustments are needed, the program TESTA.SAV interrogates the sensors and returns "TRUE" if the sensor sees the rabbit (the light will go out also). Be sure to put the sensor on a length of 1-1/8" ID polyethylene tubing so conditions while running TESTA match the rabbit system. TESTA is less convenient for adjusting sensors 1 and 2 in the RTNS-1 target pit. (Sensor 1 is closest to the target. These sensors are redundant; the user may choose either sensor for use in any program, in case one fails, and the pit is "too hot" to enter for maintenance.)

Each sensor consists of two circuit boards and the sensor itself. If you believe a sensor is bad, plug it into a board that you know is working properly. It is best to turn the multiplexor off while making any changes. Make sure ALL 26 pins are plugged into the connector. If this sensor network now fails, the sensor itself is bad, otherwise one of the circuit boards is bad. There are many spare boards in the multiplexor box. Return the sensor to the board that fails and try 2 or 3 green boards, making many sensitivity adjustments. If the sensor still fails, swap L-shaped boards (leaving no blanks in the chain) keeping the original green board, and repeat sensitivity adjustments. It is easy to misalign the 26-pin connector, so double check.

APPENDIX C: SOFTWARE

Detailed knowledge of the programs controlling the rabbit system is not needed to operate the system. This section is included in the event that changes are needed in the software. An understanding of the RT-11 operating system, especially editing and file manipulation, and the OMSI (Oregon Minicomputer Software Inc.) implementation of PASCAL is required to make any changes.

Except for *BYHAND*, all of the programs are broken into many external procedures, a list of global variables, and a main program. Procedures can be individually edited and compiled with the global variables without altering the rest of the program. The final program is obtained by linking all of the object files created by compilation of the main program and the procedures. For instructions regarding the PASCAL compiler consult the OMSI PASCAL manual.

Because PASCAL operates via a once-through compiler, all constants, variables, procedures, and functions must be defined before being used in the program. When a procedure is being individually compiled, all other procedures to be called must be declared as external with an OMSI PASCAL "EXTERNAL" statement (see the source listings of *CYCLE* following this appendix). Table 3 gives the object-file names and correct order for the system programs. File names are usually the first few letters of the procedure name, which, like all identifiers in PASCAL, can be of any length. An excellent reference, serving as a text book, is Programming in PASCAL, by Peter Grogono, which is available from the Small Computer User's Library.

TABLE 3. RABBIT SYSTEM PROGRAMS AND THEIR
CONSTITUENT PROCEDURES AND FUNCTIONS.

MANUAL	CYCLE/ACYCLE ⁺ (CYCLE6/ZCYCLE ⁺)*	CYCLES
MAIN1	MAIN2 ⁺	MAIN4
GLOB1	GLOBAL	GLOBAL
SHOWH0	SHOWH2	SHOWH2
INIT	INIT2	INIT2
	INITI2	INITI2
INITIA	IRRADI*	IRRAD6
SEE	SEE2	SEE2
SEND	SEND2	SEND2
BRING1	BRING2	BRING2
RETURN	RETUR2	RETUR2
RETRIE	RETRI2	RETRI2
	KEEPTR	
	SECOND	SECOND
	START	START

⁺MAIN3 for ACYCLE and ZCYCLE

*Detector "start-pulse" rather than gate in IRRAD6.

Source listings of all necessary program pieces are kept on floppy disks in T940, and are identified as filename.PAS. Object files of each are identified as filename.OBJ, while executable programs have the identifier .SAV. If a program change does not introduce new global variables, a newly created object file can be linked with the previously existing object files. An example of the linking procedure, for CYCLE, is as follows:

R LINK (return)

*CYCLE=MAIN2,SHOWH2,INIT2,INITI2,IRRADI,/C (return)

*SEE2,SEND2,BRING2,RETUR2,RETRI2,/C (return)

*KEEPTR,SECOND,START,PASCAL (return)

*CTRL-C

The file, CYCLE.SAV, is thus created and can be executed with the command RUN CYCLE. The prompt from the linker is "**", while: "/C" tells the linker the command is continued on the next line. PASCAL.OBJ the last object file in any link operation, is the OMSI PASCAL support library.

Finally, the source listings of the components of CYCLE are presented as an example of the system software. Comments for the purpose of documentation are enclosed in special brackets (* like this *).

1990 1991 1992 1993 1994 1995 1996 1997 1998 1999 2000 2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022 2023 2024 2025 2026 2027 2028 2029 2030 2031 2032 2033 2034 2035 2036 2037 2038 2039 2040 2041 2042 2043 2044 2045 2046 2047 2048 2049 2050 2051 2052 2053 2054 2055 2056 2057 2058 2059 2060 2061 2062 2063 2064 2065 2066 2067 2068 2069 2070 2071 2072 2073 2074 2075 2076 2077 2078 2079 2080 2081 2082 2083 2084 2085 2086 2087 2088 2089 2090 2091 2092 2093 2094 2095 2096 2097 2098 2099 2100 2101 2102 2103 2104 2105 2106 2107 2108 2109 2110 2111 2112 2113 2114 2115 2116 2117 2118 2119 2120 2121 2122 2123 2124 2125 2126 2127 2128 2129 2130 2131 2132 2133 2134 2135 2136 2137 2138 2139 2140 2141 2142 2143 2144 2145 2146 2147 2148 2149 2150 2151 2152 2153 2154 2155 2156 2157 2158 2159 2160 2161 2162 2163 2164 2165 2166 2167 2168 2169 2170 2171 2172 2173 2174 2175 2176 2177 2178 2179 2180 2181 2182 2183 2184 2185 2186 2187 2188 2189 2190 2191 2192 2193 2194 2195 2196 2197 2198 2199 2200 2201 2202 2203 2204 2205 2206 2207 2208 2209 2210 2211 2212 2213 2214 2215 2216 2217 2218 2219 2220 2221 2222 2223 2224 2225 2226 2227 2228 2229 2230 2231 2232 2233 2234 2235 2236 2237 2238 2239 2240 2241 2242 2243 2244 2245 2246 2247 2248 2249 2250 2251 2252 2253 2254 2255 2256 2257 2258 2259 2260 2261 2262 2263 2264 2265 2266 2267 2268 2269 2270 2271 2272 2273 2274 2275 2276 2277 2278 2279 2280 2281 2282 2283 2284 2285 2286 2287 2288 2289 2290 2291 2292 2293 2294 2295 2296 2297 2298 2299 2300 2301 2302 2303 2304 2305 2306 2307 2308 2309 2310 2311 2312 2313 2314 2315 2316 2317 2318 2319 2320 2321 2322 2323 2324 2325 2326 2327 2328 2329 2330 2331 2332 2333 2334 2335 2336 2337 2338 2339 2340 2341 2342 2343 2344 2345 2346 2347 2348 2349 2350 2351 2352 2353 2354 2355 2356 2357 2358 2359 2360 2361 2362 2363 2364 2365 2366 2367 2368 2369 2370 2371 2372 2373 2374 2375 2376 2377 2378 2379 2380 2381 2382 2383 2384 2385 2386 2387 2388 2389 2390 2391 2392 2393 2394 2395 2396 2397 2398 2399 2400 2401 2402 2403 2404 2405 2406 2407 2408 2409 2410 2411 2412 2413 2414 2415 2416 2417 2418 2419 2420 2421 2422 2423 2424 2425 2426 2427 2428 2429 2430 2431 2432 2433 2434 2435 2436 2437 2438 2439 2440 2441 2442 2443 2444 2445 2446 2447 2448 2449 2450 2451 2452 2453 2454 2455 2456 2457 2458 2459 2460 2461 2462 2463 2464 2465 2466 2467 2468 2469 2470 2471 2472 2473 2474 2475 2476 2477 2478 2479 2480 2481 2482 2483 2484 2485 2486 2487 2488 2489 2490 2491 2492 2493 2494 2495 2496 2497 2498 2499 2500 2501 2502 2503 2504 2505 2506 2507 2508 2509 2510 2511 2512 2513 2514 2515 2516 2517 2518 2519 2520 2521 2522 2523 2524 2525 2526 2527 2528 2529 2530 2531 2532 2533 2534 2535 2536 2537 2538 2539 2540 2541 2542 2543 2544 2545 2546 2547 2548 2549 2550 2551 2552 2553 2554 2555 2556 2557 2558 2559 2560 2561 2562 2563 2564 2565 2566 2567 2568 2569 2570 2571 2572 2573 2574 2575 2576 2577 2578 2579 2580 2581 2582 2583 2584 2585 2586 2587 2588 2589 2590 2591 2592 2593 2594 2595 2596 2597 2598 2599 2600 2601 2602 2603 2604 2605 2606 2607 2608 2609 2610 2611 2612 2613 2614 2615 2616 2617 2618 2619 2620 2621 2622 2623 2624 2625 2626 2627 2628 2629 2630 2631 2632 2633 2634 2635 2636 2637 2638 2639 2640 2641 2642 2643 2644 2645 2646 2647 2648 2649 2650 2651 2652 2653 2654 2655 2656 2657 2658 2659 2660 2661 2662 2663 2664 2665 2666 2667 2668 2669 2670 2671 2672 2673 2674 2675 2676 2677 2678 2679 2680 2681 2682 2683 2684 2685 2686 2687 2688 2689 2690 2691 2692 2693 2694 2695 2696 2697 2698 2699 2700 2701 2702 2703 2704 2705 2706 2707 2708 2709 2710 2711 2712 2713 2714 2715 2716 2717 2718 2719 2720 2721 2722 2723 2724 2725 2726 2727 2728 2729 2730 2731 2732 2733 2734 2735 2736 2737 2738 2739 2740 2741 2742 2743 2744 2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766 2767 2768 2769 2770 2771 2772 2773 2774 2775 2776 2777 2778 2779 2780 2781 2782 2783 2784 2785 2786 2787 2788 2789 2790 2791 2792 2793 2794 2795 2796 2797 2798 2799 2800 2801 2802 2803 2804 2805 2806 2807 2808

```

17  FILL OF ALL CONSTANT AND VARIABLES DEFINED OVER ALL OF THE PROGRAM *)
18  'CLOCK', MUST BE SELF EXPLANATORY. TIMES ARE LISTED IN HOURS FOR *)
19  USE WITH THE DMS) PARAM) FLOATING-POINT LINE (CLOCK, *)

```

```

14 THE "PUSH" COMMAND ALLOWS THE PROGRAM TO CHANGE THE VALUE OF A *)
14 PARTICULAR MEMORY LOCATION. ANY NUMBER FOLLOWED BY 'B' IS AN *)
14 16-BIT NUMBER. *)

```

```

# THE CONSTANTS ARE AS FOLLOWS: *)
# INTERGATEL - NUMBER OF INTERROGATIONS IN 5 SECONDS *)
#              (DEFAULT 1 AIR SHUT-OFF TIME) *)
# FLINGDELAY - TIME (H) BETWEEN DETECTION OF RABBIT *)
#              BY SENSOR 3 AND PLUNGER DOWN *)
# DROPDDELAY - TIME (H) BETWEEN DETECTION OF RABBIT *)
#              BY SENSOR 4 AND OPEN DROPOUT *)
# SNIPT - SENSOR NUMBER IN TARGET PIT *)
# (NIDELAT - TIME (H) BETWEEN IRRAD AND COUNT *)

```

[illegible]

```

INTERPOLATE      = 5.82;          PLUNGDELAY      = 5.555E-5;
DECELDELTA      = 1.111E-3;      SEEPIT        = 1;

```

(4) IRRADIATION CYCLE DEFAULT PARAMETERS (4)

```

NUCLEOTIDES      = 10;          LENLENGTH      = 1.3888E-3;
TOTAL LENGTH     = 8.333E-3;    CNTLENGTH     = 8.333E-3;
GC              = 5.555E-4;     MAXLEN       = 70;

```

NAME VALUE CONTROL CONSTANTS FOR PRU-11 *)

```

CPLTIT      = 0B;      SENDOFF      = 100000B;
RETURNPON   = 10B;     RETURNOFF     = 100010B;
IRRADION    = 20B;     IRRADOFF      = 100020B;
BROFITT     = 40B;     CLOSEIT       = 100040B;
PLUNGDOWN   = 50B;     PLUNGUP       = 100050B;
BROFOUT     = 60B;     DETECTOR      = 100060B;
GATEOPEN    = 70B;     GATECLOSE     = 100070B;

```

```
TYPE      COL      = 1..MAXLEN;
LINE      = ARRAY (COL) OF CHAR;
TABLE     = FILE OF LINE;
```

```

FOR COUNT,SEECYCLES,SENSOR,PITSensor,PLUNGSensor,
CYCLENUM,1HOUR,1MIN,1SEC,TENTHS,CYCLES : INTEGER;

```

```
FLUNGTIME,T1,T2,DROPTIME,IRRADLENGTH,  
PAIT4NEXT,WAITING,IRRADBEGIN,IRRADEND,  
COUNTBEGIN,COUNTEND,COUNTDELAY,COUNTLENGTH      : REAL ;
```

INDEX : TABLES

```

      REPLY          CHAR;

      CONTINUE, SENDOK, RETURNOK, ALLOK, ACK, ATCOUNT      : BOOLEAN;

      RABBITID       : ARRAY [ 1..30 ] OF CHAR;

      VALVE0 ORIGIN 166660B,
      VALVE1 ORIGIN 166660B,
      VALVE2 ORIGIN 166660B,
      VALVE4 ORIGIN 166660B,
      VALVE5 ORIGIN 166660B,
      VALVE6 ORIGIN 166660B,
      GATE   ORIGIN 166660B   : INTEGER;

```

```

      (*      CYCLE TO CYCLE      *)

```

```

(*PROCEDURE SHOWHOW*)

```

```

BEGIN                                (*      SHOWHOW      *)

```

```

WRITELN('      PROGRAMS CYCLE AND ACYCLE PRESET THE IRRADIATION AND COUNTING');
WRITELN('PARAMETERS SO THAT THE LSI-11 CYCLES THE RABBIT SYSTEM.  ACYCLE');
WRITELN('REQUIRES A START SIGNAL FROM THE USER FOR EVERY CYCLE WHEREAS CYCLE');
WRITELN('WILL AUTOMATICALLY STEP THROUGH A PRESET NUMBER OF CYCLES.  BOTH');
WRITELN('CREATE A FILE, FILE.TXT, RECORDING FOR EACH CYCLE THE IRRADIATION');
WRITELN('AND COUNTING TIMES, AND A MESSAGE, "OK" OR "NO DATA", INDICATING');
WRITELN('WHETHER OR NOT THE CYCLE WAS COMPLETED SUCCESSFULLY.  THIS FILE MAY');
WRITELN('BE INSPECTED AFTER PROGRAM EXECUTION ON THE TERMINAL WITH THE ');
WRITELN('COMMAND TYPE FILE.TXT OR IT CAN BE PRINTED ON THE TELETYPE WITH THE');
WRITELN('COMMAND PRINT FILE.TXT.  THIS FILE IS DESTROYED WITH THE NEXT');
WRITELN('EXECUTION OF EITHER PROGRAM ');
WRITELN('
DURING THE OPERATION OF CYCLE, FAILURE OF ANY OF THE SENSORS');
WRITELN('TO SEE THE RABBIT WILL CAUSE NO DATA TO BE TAKEN (VIA THE GATE)');
WRITELN('AND WILL INTERRUPT THE PROGRAM.  THE USER MAY THEN DECIDE TO STOP');
WRITELN('THE RUN OR TO CONTINUE.  IF THE PROGRAM IS ABORTED AT THIS POINT,');
WRITELN('FILE.TXT WILL CONTAIN RECORDS OF THE IRRADIATIONS COMPLETED. ');
WRITELN('
MORE INFORMATION ON THE RABBIT SYSTEM SOFTWARE IS CONTAINED');
WRITELN('IN FILE RABBIT.TXT. ');
WRITELN('
WRITE('WHEN READY TO CONTINUE, TYPE "RET": ');
READLN; READ(REPLY);

```

```

END;                                (*      SHOWHOW      *)

```

```

*      FILE : INIT.PAS      *)

*      (**$+  EXTERNAL PROCEDURE *)
PROCEDURE INIT;

*      (* A VERY SHORT PROCEDURE TO INITIALIZE GLOBAL VARIABLES THAT ARE *)
*      (* RABBIT-SYSTEM OR IRRADIATION PARAMETERS TO DEFAULT VALUES LISTED *)
*      (* AS CONSTANTS, EXISTS MAINLY TO FACILITATE CHANGES IN DEFAULT   *)
*      (* VALUES IF THEY ARE EVER NEEDED.      *)

BEGIN                                (*  INIT  *)

    PLUNGTIME := PLUNGDELAY;
    DROPYIME := DROPDELAY;
    PITSENSOR := SEEPIT;
    SEECYCLES := INTERROGATE;
    CYCLES := NUMCYCLES;
    IRRADLENGTH := IRRLENGTH;
    COUNTDELAY := CNTDELAY;
    COUNTLENGTH := CNTLENGTH;
    WAIT4NEXT := WAIT

END;                                (*  INIT  *)

```

```

(*)      FILE :  INITIA.PAS      *)

(**E#  EXTERNAL PROCEDURE *)

PROCEDURE INITIALIZE;

  (* PRESENTS RABBIT-SYSTEM PARAMETERS TO THE USER AND ALLOWS CHANGES *)
  (* FOR THIS EXECUTION OF 'CYCLE'.  THE USER MAY SPECIFY: 1) THE TIME *)
  (* DELAYS BETWEEN THE 'RABBIT-HAS-RETURNED' SIGNAL AND ACTIVATION OF *)
  (* THE PLUNGER AND THE OPENING OF THE DROPOUT, 2) THE DEFAULT AIR- *)
  (* PRESSURE SHUT-OFF TIME IN CASE A RABBIT IS NOT SEEN, AND *)
  (* 3) WHICH SENSOR TO BE USED AT THE TARGET END. *)

CONST   SECS      = 3600;
        CYCLETIME = 8.8E-4;      (* TIME (S) NEEDED FOR ONE INTERROGATION *)
                                      (* OF A SENSOR *)

VAR     CHOICE    : CHAR;
        D,P,T    : REAL;
        S        : INTEGER;
        LIST,CHANGES : BOOLEAN;

BEGIN                                     (* INITIALIZE *)

  I := SEECYCLES * CYCLETIME;
  P := PLUNGTIME * SECS;
  D := DROPTIME * SECS;
  S := PITSENSOR;

  WRITELN;WRITELN;

  REPEAT
    WRITELN('THE ADJUSTABLE PARAMETERS, THEIR VALUES, AND ONE-LETTER CODES',
            ' ARE:');
    WRITELN;
    WRITELN('PLUNGER DELAY TIME (P) = ',P:7:3, ' SEC');
    WRITELN('DROPOUT DELAY TIME (D) = ',D:5:1, ' SEC');
    WRITELN('DEFAULT SHUT-OFF (T) = ',T:5:1, ' SEC');
    WRITELN('PIT SENSOR 1 OR 2 (S) = ',S:3);

    LIST := FALSE;
    CHANGES := TRUE;

    REPEAT
      WRITELN;
      WRITELN('TYPE ONE-LETTER CODE (P,D,T,S) TO CHANGE A PARAMETER,');
      WRITE ('L' TO LIST, 'X' TO EXIT, 'N' TO EXIT WITHOUT CHANGES:');
      READLN;READ(CHOICE);
      CASE CHOICE OF
        'D' : BEGIN
          WRITELN;
          WRITE('DROPOUT DELAY TIME (D) = ',D:5:1, ' SEC');
          READLN;READ(D);
          END;

        'S' : BEGIN
          WRITELN;
          WRITE('PIT SENSOR IS NOW (S) = ',S:3, ' ');
          READLN;READ(S);
          END;
      END;
    UNTIL CHOICE = 'N' OR CHOICE = 'X';

  WRITELN;

```

```

1' : BEGIN
    WRITELN;
    WRITE('DEFAULT AIR SHUT-OFF TIME (T) = ',T:5:1,' : ');
    READLN;READ(T);
END;

'P' : BEGIN
    WRITELN;
    WRITELN(' ***** CAUTION *****');
    WRITELN('CHANGING P EFFECTS RABBIT SYSTEM PERFORMANCE!');
    WRITELN('BEST VALUE IS :',PLUNGTIME*SECS:7:3,' SEC');
    WRITE('PLUNGER DELAY TIME (P) = ',P:7:3,' : ');
    READLN;READ(P);
END;

'L' : BEGIN
    CHANGES := FALSE;
    LIST := TRUE;
END;

'N' : EXIT;

'X' : CHANGES := FALSE;

ELSE CHANGES := TRUE
END
(* CASES *)
UNTIL CHANGES = FALSE;

UNTIL LIST = FALSE;
IF CHOICE = 'X' THEN
    BEGIN
        PLUNGTIME := P/SECS;
        DROPTIME := D/SECS;
        PITSENSOR := S;
        SEECYCLES := TRUNC ( T/CYCLETIME );
    END;
    (* UPDATE *)
END;
(* INITIALIZE *)

```



```

(*) FILE : IRRADI.PAS      *)

(**E# EXTERNAL TO CYCLE *)

PROCEDURE IRRADINIT;

(* ALLOWS THE USER TO SET IRRADIATION/COUNTING PARAMETERS FOR A SERIES *)
(* OF CYCLIC ACTIVATIONS: 1) NUMBER OF CYCLES, 2) IRRADIATION LENGTH, *)
(* 3) DELAY BEFORE COUNTING, 4) COUNT LENGTH, AND 5) WAIT PERIOD *)
(* BEFORE THE NEXT CYCLE. *)

CONST SECS = 3600;
      CYCLETIME = 8.8E-4;

VAR CHOICE : CHAR;
    CHOOSF,LIST : BOOLEAN;
    I,D,W,C : REAL;
    N : INTEGER;

BEGIN                                (* IRRADINIT *)

    (* INITIALIZE TO DEFAULT VALUES *)

    N := CYCLES;
    I := IRRADLENGTH * SECS;
    D := COUNTDELAY * SECS;
    C := COUNTLENGTH * SECS;
    W := WAIT4NEXT * SECS;

    (* LIST CURRENT IRRADIATION PARAMETERS *)

    REPEAT                            (* UNTIL LIST = FALSE *)

        WRITELN;WRITELN;
        WRITELN('THE IRRADIATION-CYCLE PARAMETERS,THEIR VALUES, AND THEIR ONE');
        WRITELN('LETTER ACCESS CODES ARE:');
        WRITELN;
        WRITELN('NUMBER OF CYCLES (N) = ',N:4);
        WRITELN('IRRADIATION LENGTH (I) = ',I:5:1,' SEC');
        WRITELN('DELAY BEFORE COUNT (D) = ',D:5:1,' SEC');
        WRITELN('COUNT LENGTH (REAL TIME) (C) = ',C:5:1,' SEC');
        WRITELN('WAIT BETWEEN CYCLES (W) = ',W:5:1,' SEC');
        WRITELN;WRITELN;

        LIST := FALSE;
        CHOOSE := TRUE;

        REPEAT                        (* MAKE CHANGES *)

            WRITELN;WRITELN;
            WRITELN('TYPE ONE-LETTER CODE (N,I,D,C, OR W) TO CHANGE A PARAMETER,');
            WRITELN('L TO LIST PRESENT VALUES, X TO EXIT, OR N ',
                'TO EXIT WITHOUT CHANGES.');
            WRITE('ENTER OPTION:');
            READLN;READ(CHOICE);
            WRITELN;WRITELN;

            CASE CHOICE OF

                'N' : BEGIN

```

```

        WRITELN('NUMBER OF CYCLES (M) = ',M:4,' : ');
        READLN#READ(M);
        WRITELN(' M:4,' CYCLES')
    END;

    'I' : BEGIN
        WRITELN('IRRADIATION LENGTH (I) = ',I:5:1,' : ');
        READLN#READ(I);
        WRITELN(' I:5:1,' SEC')
    END;

    'D' : BEGIN
        WRITELN('DELAY BEFORE COUNT (D) = ',D:5:1,' : ');
        READLN#READ(D);
        WRITELN(' D:5:1,' SEC )
    END;

    'C' : BEGIN
        WRITELN('COUNT LENGTH (C) = ',C:5:1,' : ');
        READLN#READ(C);
        WRITELN(' C:5:1,' SEC')
    END;

    'W' : BEGIN
        WRITELN('WAIT BETWEEN CYCLES (W) = ',W:5:1,' : ');
        READLN#READ(W);
        WRITELN(' W:5:1,' SEC')
    END;

    'L' : BEGIN
        CHOOSE := FALSE;
        LIST := TRUE;
    END;

    'N' : EXIT;

    'X' : CHOOSE := FALSE;

    ELSE CHOOSE := TRUE

    END
    (* CASES *)
    UNTIL CHOOSE = FALSE

    UNTIL LIST = FALSE;

    IF CHOICE = 'X' THEN
        BEGIN
            CYCLES := M;
            IRRADLENGTH := I/SECS;
            COUNTDELAY := D/SECS;
            COUNTLENGTH := C/SECS;
            WAITANEXT := W/SECS;
            END;
            (* UPDATED *)
        END;
        (* IRRADINIT *)
    
```

```

(*) FILE : SEL.PAS      *)

(**+ EXTERNAL FUNCTION **)
FUNCTION SEE ( SENSOR : INTE... ) : BOOLEAN;

(* "FUNCTION SEE" INTERROGATES A SENSOR, SPECIFIED WHEN SEE IS CALLED, *)
(* UNTIL IT SEES A RABBIT OR REACHES A PRESET NUMBER OF INTERROGATIONS. *)
(* "SEE" HAS A BOOLEAN VALUE; IT RETURNS "TRUE" OR "FALSE" TO THE *)
(* POINT FROM WHICH IT WAS CALLED. *)

(* THE DEFAULT AIR-PRESSURE TIME LIMIT IS SET BY SPECIFYING THE NUMBER *)
(* OF INTERROGATIONS BEFORE SEE RETURNS "FALSE". A FIVE SECOND DEFAULT *)
(* REQUIRES 56,820 INTERROGATIONS. SINCE THIS IS GREATER THAN THE *)
(* ALLOWED MAXIMUM, THE INTERROGATE COMMAND IS NESTED FURTHER IN A *)
(* DECAY LOOP. THE ACTUAL TIME ELAPSED IN ONE INTERROGATION IS ABOUT *)
(* 88 MICROSECONDS. *)

(* A DRV-11 BOARD PROVIDES THE INTERFACE BETWEEN THE LSI-11 AND THE *)
(* SENSOR MULTIPLEXOR. DOCUMENTATION CONCERNING THE USE OF THIS *)
(* SYSTEM IN NURE, INCLUDING PROGRAMS IN MACRO-11, IS ON A DISK AT *)
(* T-940 IN THE FILE, FILE.MAC, AND FURTHER INFORMATION IS INCLUDED *)
(* IN CORE-81773. "ARITHMETIC CIRCUIT ELIMINATES ADDRESS JUMPERS AND *)
(* DIP SWITCHES" BY RICHARD JOSSE. CONSULT THE LSI-11, FRP-11/03 *)
(* USER'S MANUAL FOR A DESCRIPTION OF THE DRV-11. *)

(* SEE FIRST DISABLES THE INTERRUPT. THEN, (256 - SENSOR NUMBER) IS *)
(* LOADED IN THE DRV-11 OUTPUT, SEEOUT, AND THE ZERO-ENABLE IS OFF. *)
(* A CARRY FROM THE 3-BIT ADDRESS OF SEEOUT SPECIFIES WHICH SENSOR IS *)
(* BEING INTERROGATED). THE DRV-11 INPUT IS EQUAL TO "SIGNAL" ( "B" *)
(* IMPLIES OCTAL ) IF A RABBIT BREAKS THE IR-BEAM TO THE SENSOR. *)

CONST  INRPT  = 1000;
        ZEROEN = 200000;
        SIGNAL = 1000000;

VAR     SEECR  ORIGIN 14710R;
        SEEOUT ORIGIN 14712R;
        SEEIN  ORIGIN 14714R : INTEGER;
        SAW    : BOOLEAN;

        DECADE, INDEX      : INTEGER;

BEGIN
    (* FUNCTION SEE *)
    SEE := FALSE;
    SEECR := SEECR AND NOT INRPT;
    SEEOUT := ((SEEOUT AND NOT 377B) + ( 256 - SENSOR )) AND NOT ZEROEN;
    FOR DECADE := 1 TO 10 DO
        BEGIN
            FOR INDEX := 1 TO SEECYCLES DO
                IF ( SEEIN AND SIGNAL ) = 0
                    THEN SEE := FALSE;
                ELSE BEGIN SEE := TRUE; SAW := TRUE; EXIT END; (* SAW RABBIT *)
            IF SAW = TRUE THEN EXIT END;
        END;
    END;
    (* FUNCTION SEE *)

```

```

(*      CALL SEND,PAS      *)

(*----- EXTERNAL PROCEDURE -----*)

FUNCTION SEE(SENSOR : INTEGER) : BOOLEAN; EXTERNAL;

PROCEDURE SEND;

(* 'SEND' OPENS VALVE 1 TO SEND THE RABBIT TO THE ICI AND THEN CALLS *)
(* 'SEE'. IF 'SEE' RETURNS 'TRUE', VALVE A IS CLOSED, VALVE E IS *)
(* OPENED TO HOLD THE RABBIT IN PLACE, AND THE MESSAGE, 'RABBIT *)
(* AT PIT' IS DISPLAYED. IF 'SEE' RETURNS 'FALSE' AFTER THE DEFAULT *)
(* TIME HAS ELAPSED, VALVE A IS CLOSED AND A WARNING GIVEN. *)

VAR      ATPIT : BOOLEAN;

BEGIN

(*      SEND      *)

  SENSOR := PITSSENSOR;
  VALVE1 := FLUPOUP;
  VALVE2 := SENDIT;
  ATPIT := SEE (SENSOR);
  IRRADREDIN := TIME;
  VALVE0 := SENDOFF;

  IF ATPIT THEN
    BEGIN
      (*      SAW RABBIT AT PIT      *)
      (*      VALVE2 HOLDS RABBIT UP TO TARGET      *)
      VALVE2 := IRRADON;
      SENSOR := TRUE;
      WRITELN(WRITELN('RABBIT AT TARGET. '))
    END
  ELSE
    BEGIN
      (*      BAD NEWS      *)
      SENSOR := FALSE;
      WRITELN;
      WRITELN('*****');
      WRITELN('***                               ***');
      WRITELN('***      RABBIT NOT AT IRRADIATION STATION !      ***');
      WRITELN('***                               ***');
      WRITELN('*****');
      WRITELN;
    END
  END;

(*      BAD NEWS      *)

END;

(*      SEND      *)

```

```

      VALVE 1 := BRIDGEUP;
    CALL 1 INTERNAL INTERRUPT;
    COUNTDOWN (SENSOR := INTERNAL) := BRIDGE+EXTERNAL;

  PROCEDURE BRIDGETRACK := SENSOR := INTERNAL;

  IF "BRIDGETRACK" CLOSE VALVE 1, OPEN VALVE 2 TO BRING THE RABBIT *)
  IF RABBIT TO 1 SAY, AND CHECK "SET". THE PLUNGER IS ACTIVATED AFTER A *)
  IF DEFEAT OF "PLUNGER" FOLLOWING THE CLOSING OF VALVE 2. *)

  GO TO ALAB := BRIDGETRACK;

  BEGIN
    (* BRINGITBACK *)
    VALVE 2 := INTERNAL;
    VALVE 1 := RETURNOFF;
    COUNTDOWN := SET (SENSOR);

    IF NOT ALAB
      GOTO
      BEGIN
        RETURNOFF := FALSE;
        UPTELETYPE;
        WRITELN(*****);
        SETTELETYPE;
        UPTELETYPE RABBIT NOT AT TRAILER;
        UPTELETYPE;
        WRITELN(*****);
        UPTELETYPE;

        IF
          (* BAD NEWS *)

        CLOSE RETURNOFF := TRUE;

        VALVE 1 := RETURNOFF;
        T1 := TIME;
        REPEAT
          T2 := TIME
        UNTIL T2 - T1 > PLUNGETIME;
        VALVE 2 := PLUNGEDOWN;
      END;
    (* BRINGITBACK *)

```



```

(*)      TITLE : RETRIE.PAL      *)

(**E+  EXTERNAL TO CYCLE *)
PROCEDURE BRINGITBACK ( SENSOR : INTEGER ) ; EXTERNAL;
FUNCTION SEE (SENSOR : INTEGER ) : BOOLEAN; EXTERNAL;

PROCEDURE RETRIEVE;

(* 'RETRIEVE' BRINGS A RABBIT TO THE DROPOUT TO REMOVE IT FROM THE *)
(* SYSTEM.  IT SENDS THE RABBIT, OPENS THE SWITCH TO THE DROPOUT, *)
(* CALLS 'BRINGITBACK', AND DROPS THE RABBIT INTO A LEAD PIG.  *)

CONST    PAUSE1  = 1E-4;
          PAUSE2  = 1E-3;

VAR      T3,T4,T5,T6,T7      : REAL;
          ATPIT    : BOOLEAN;

BEGIN
  SENSOR := 4;
  VALVE0 := SENDIT;
  ATPIT := SEE (PITSENSOR);
  T3 := TIME;
  VALVE0 := SENDOFF;
  VALVE6 := DROPOUT;
  REPEAT
    T4 := TIME
  UNTIL T4 - T3 > PAUSE1;
  BRINGITBACK (SENSOR);
  T5 := TIME;
  REPEAT
    T6 := TIME
  UNTIL T6 - T5 > DROPTIME;
  VALVE4 := DROFIT;
  VALVE6 := DETECTOR;
  REPEAT
    T7 := TIME
  UNTIL T7 - T6 > PAUSE2;
  VALVE4 := CLOSEIT;
  VALVE5 := PLUNGUP;

END;                                     (* RETRIEVE *)

```

```

(* FILE : SECOND.PAS *)

(*#E+ PROCEDURE EXTERNAL TO CYCLE *)

PROCEDURE SECONDS ( VAR T : REAL );

(* FOR EACH TIME ORIGINALLY RECORDED IN DECIMAL HOURS, 'SECONDS' *)
(* RETURNS HOURS, MINUTES, SECONDS, AND TENTHS OF SECONDS. *)

VAR RMIN,RSEC : REAL;

BEGIN
    (* SECONDS *)
    IHOUR := TRUNC(T);
    RMIN := (T - IHOUR) * 60;
    IMIN := TRUNC(RMIN);
    RSEC := (RMIN - IMIN) * 60;
    ISEC := TRUNC(RSEC);
    TENTHS := ROUND ((RSEC - ISEC) * 10 );

    IF TENTHS = 10 THEN BEGIN ISEC := ISEC + 1; TENTHS := 0;END;
    IF ISEC = 60 THEN BEGIN IMIN := IMIN + 1; ISEC := 00;END;
    IF IMIN = 60 THEN BEGIN IHOUR:=IHOUR+1; IMIN := 00; END

END;
    (* SECONDS *)

```



```

(*) FILE : KEEPTR.PAS      *)

(**E+ PROCEDURE EXTERNAL TO CYCLE *)
PROCEDURE SECONDS ( VAR T : REAL ); EXTERNAL;

PROCEDURE KEEPTRACK;

(** SETS UP 'FIL.TXT' TO STORE TIMES FOR EACH CYCLE AND WRITES THE *)
(** TIMES AFTER EACH CYCLE. *)

VAR
    CYCLENUM,I,J      : INTEGER;
    T                : ARRAY 11..47 OF REAL;
    ANS              : CHAR;

BEGIN
    (* KEEPTRACK *)

    IF COUNT = 1 THEN BEGIN
        (* INITIALIZE FILE *)
        WRITELN(INFILE,' :18,'RECORD OF IRRADIATIONS');
        WRITELN(INFILE);WRITELN(INFILE);
        WRITELN(INFILE,'CYCLE   IRRAD START   IRRAD END   COUNT START ',
        'COUNT END   MESSAGE');
        WRITELN(INFILE);
    END;
    (* INITIALIZATION *)

    CYCLENUM := COUNT;
    T(1) := IRRADBEGIN;
    T(2) := IRRADEND;
    T(3) := COUNTBEGIN;
    T(4) := COUNTEND;
    WRITE(INFILE,CYCLENUM : 5);
    FOR J:= 1 TO 4 DO
        BEGIN
            (* WRITE THE TIME *)
            SECONDS (T(J));
            WRITE(INFILE,' :5,1HOUR:2,':',1MIN:2,':',1SEC:2,'.',TENTHS:1);
        END;
        (* WRITING TIMES *)

    IF ALLON THEN WRITELN(INFILE,' OK')
    ELSE WRITELN(INFILE,' NO DATA');
    (* KEEPTRACK *)
END;

```

```

1      1111 : START.PAS      *)

(*--- EXTERNAL TO CYCLE ---*)
PROCEDURE SHOWHOW; EXTERNAL;
PROCEDURE INIT; EXTERNAL;
PROCEDURE INITIALIZE; EXTERNAL;
PROCEDURE IRRADINIT; EXTERNAL;

* PROCEDURE START;

(* *START* IS HALF OF THE MAIN PROGRAM, CALLING INITIALIZATION *)
(* PROCEDURES AND GIVING INSTRUCTIONS. *)

VAR      PRELIMINARIES      : BOOLEAN;

BEGIN                                     (* START *)

    WRITE('DO YOU NEED INSTRUCTIONS? Y OR N(RETURN) :');
    READLN#READ(REPLY);
    IF REPLY = 'Y' THEN SHOWHOW;

    WRITE#WRITE('NEED TO INITIALIZE PARAMETERS? Y OR N(RET) :');
    READLN;
    READ#READ(REPLY);
    INIT;
    IF REPLY = 'Y' THEN INITIALIZE;

    IRRADINIT;
    PRELIMINARIES := FALSE;

    WHILE PRELIMINARIES = FALSE DO
        BEGIN                                (* LOOP TO GET READY *)

            WRITELN#WRITELN;
            WRITELN('WHEN READY TO START THE IRRADIATION CYCLE TYPE "RET";');
            WRITELN('OTHERWISE, TYPE "H" FOR HELP, "N" TO ABORT THE RUN,');
            WRITELN(' "I" TO INITIALIZE THE RABBIT SYSTEM PARAMETERS, OR');
            WRITELN(' "C" TO INITIALIZE IRRADIATION-CYCLE PARAMETERS. ');
            WRITELN#WRITE('ENTER OPTION: ');
            READLN#READ(REPLY);
            CASE REPLY OF

                'H' : PRELIMINARIES := TRUE;
                'I' : SHOWHOW;
                'C' : INITIALIZE;
                'I' : IRRADINIT;
                'N' : WRITE('TO ABORT THE RUN, TYPE "CTRL-C":');
            ELSE PRELIMINARIES := FALSE;

            END#
        END#                                (* CASES *)
        (* PRELIMINARIES LOOP *)
    END#

END#                                     (* START *)

```

```
(* FILE : MAIN.PAS *)
```

```

PROCEDURE START; EXTERNAL;
PROCEDURE SHOWHOW; EXTERNAL;
PROCEDURE INIT; EXTERNAL;
PROCEDURE INITIALIZE; EXTERNAL;
PROCEDURE IRRADINIT; EXTERNAL;
FUNCTION SEE ( SENSOR : INTEGER ) : BOOLEAN; EXTERNAL;
PROCEDURE SEND; EXTERNAL;
PROCEDURE BRINGOTBACK ( SENSOR : INTEGER ); EXTERNAL;
PROCEDURE RETURN; EXTERNAL;
PROCEDURE SECONDS ( VAR T : REAL ); EXTERNAL;
PROCEDURE KEEPTACK; EXTERNAL;
PROCEDURE RETRIEVE; EXTERNAL;

```

```

(* AFTER 'MAIN' CALLS 'START', IT BEGINS THE CYCLE LOOP, COUNTING *)
(* THE CYCLES. IT CALLS 'RETURN' AND IF EVERYTHING WAS OK, DISPLAYS *)
(* THE RETURN TIME, AND STARTS THE DETECTOR. ANY ERRORS REQUIRE THE *)
(* USER TO RESTART THE SERIES OR TO ABORT. 'MAIN' TIMES THE COUNT *)
(* AND THE WAIT BETWEEN CYCLES, AND THEN CALLS 'KEEPTACK'. *)

```

```

BEGIN
COUNT := 0;
START;
CONTINUE := TRUE;
REWRITE(INFILE, 'DK:FILE.TXT');
WHILE CONTINUE DO

```

```

BEGIN

```

```

RETURN;
COUNT := COUNT + 1;
PLUMBSSENSOR := 5;
SENSOR := PLUMBSSENSOR;
T2 := ( T2 - IRRADEND ) * 3600;
ATCOUNT := SEE (SENSOR);
IF (RETURNOK AND ATCOUNT) THEN
WRITELN('RABBIT AT DETECTOR. TIME = ', T2:3:1, ' SECS');

```

```

ALLOW := AOK AND ATCOUNT;
IF ALLOW

```

```

THEN
BEGIN
REPEAT
COUNTBEGIN := TIME
UNTIL (COUNTBEGIN - IRRADEND) > COUNTDELAY;
GATE := GATEOPEN;
WRITELN('CYCLE = ', COUNT:3, ' ALL OK. ');
END

```

```

ELSE
BEGIN
WRITELN;WRITELN;
WRITE('TO RESTART, TYPE "RET", TO ABORT TYPE "N": ');
READLN;READ(REPLY);
IF REPLY = 'N' THEN EXIT;
END;

```

```

REPEAT
    COUNTEND := TIME
    UNTIL ( COUNTEND - COUNTBEGIN ) > COUNTLENGTH;
    GATE := GATECLOSE;          (* STOP COUNTING *)
    KEEPTIME;                    (* STORE TIME INFORMATION *)
REPEAT
    WAITING := TIME
    UNTIL ( WAITING - COUNTEND ) > WAIT4NEXT;      (* AGAIN *)
    IF COUNT = CYCLES THEN EXIT;    (* ALL DONE! *)
END;                               (* CYCLE LOOP *)

VALUES := PLUNGUP;
WRITE('DO YOU NEED TO RETRIEVE RABBIT? Y OR N(RET): ');
READLN;READ(REPLY);
IF REPLY = 'Y' THEN RETRIEVE;
WRITELN (' COUNT =',COUNT);
CLOSE(INFILE)

END.                               (* MAIN *)

```

Figure 1. Rabbit loader and plunger. A rabbit is placed in the loader (circular section) with the sample facing to the left, and then sent to the RTNS-1. If it is to be counted, it is returned to the plunger at the end of the line and positioned in front of the detector. An infra-red sensor is shown on the far left.

Figure 2. The irradiation station adjacent to the RTNS-I target assembly.

Figure 3. Schematic of the rabbit system showing the location of major system components, valves (lettered), and sensors (S1-5).

Figure 4. Rabbit body, sample capsule, and snap-in cap. The scale pictured is calibrated in cm.

Figure 5. Details of the 14-MeV neutron irradiation geometry utilizing the RTNS-1 rabbit system (1 1/2 times actual size).