

# All You've Ever Wanted In a Relay or Switch Tester.

**B.J.Frost CEng MIEE**

Managing Director, Applied Relay Testing Ltd, England

**Abstract** – Applied Relay Testing Ltd is a specialist Company that has created a range of advanced test products dedicated to relays and switch devices which cover the test areas of low-level parametric measurement, contact life-testing and high-voltage performance.

This paper will debut an entirely new concept in testing electronic components by using low-cost modules pre-configured for your common test requirements whether this be relay parametric testing, relay life-testing or the test of other non-relay parts. A working demonstration will be made of this technique implementing a low-cost relay characterisation system that measures and records contact parameters and timing measurements routinely on every cycle during a life-test.

Other examples will illustrate how you can build a test system to your own requirements or to simply enhance your existing equipment without the need to 'start over'.

## I. INTRODUCTION.

It started with an idea for a flexible life-test system [2]. Our company had first produced life-test equipment for telecom relays back in the mid 90's with our RT96, a test system that quickly established itself within the low to mid-sized relay power band and with very specific telecom resistive, cable and inductive loads [1]. More and more often now though, in addition to conventional low power and signal relay life testing we find that we are being asked for high-power relay and switch testing, but at a much lower cost. With the cost of the humble PC continually falling, everyone expects to pay less and less .....

We decided that we needed a way of making a life-test system that met the following main design goals:

- Uncomplicated.
- Minimal hardware.
- Operates over a wide cycle rate range.
- Copes with high or low power devices.
- Measures contact resistance & stick voltage
- Tests against customer limits.
- Produces a report.

The high-power capability would be most useful to us to cope with increasing customer demands for automotive loads, safety relays and other high voltage or current life-test applications, yet we realised that customers should

only be asked to pay for capability that they needed. What was required was a 'Lego' style of life-test system construction made up of building blocks that were small enough to create very tightly focussed test systems and hence able to deliver a capability tied closely to its cost.

We started by sketching out a typical circuit of a high power 12V automotive life-test on a single contact as showing in Figure 1 below.

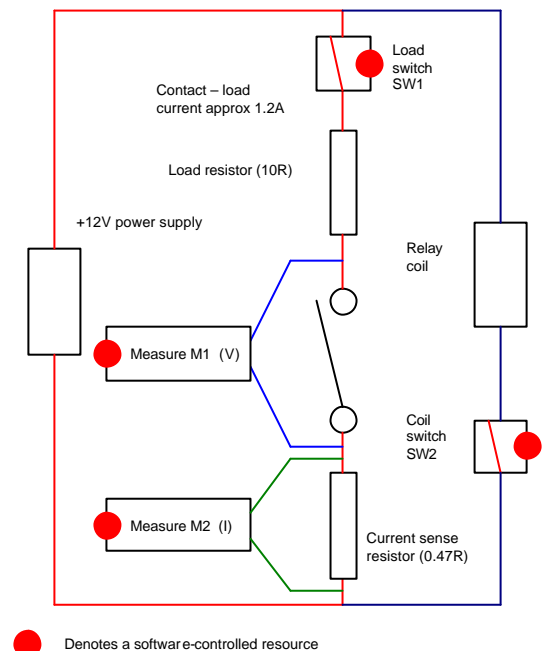


Figure 1 Simple high-power life-test circuit, e.g. automotive.

The key points are:

- The relay is enabled and disabled by a semiconductor switch SW2
- The load can be enabled or disabled by semiconductor switch SW1
- Contact voltage is measured by a voltage measurement input channel M1
- Contact current is measured by a voltage measurement input channel M2.

We quickly realised that we were building the circuit using very standard electrical building block resources – voltage measurement inputs and semiconductor switches -

and that given enough of both, we could make pretty much any measurement on any relay, even parametric measurements such as operate and release voltage or timing measurements.

Our fingers were itching so we built a ‘concept vehicle’ – the Reflex51 (mark 1) shown in Figure 2 below.

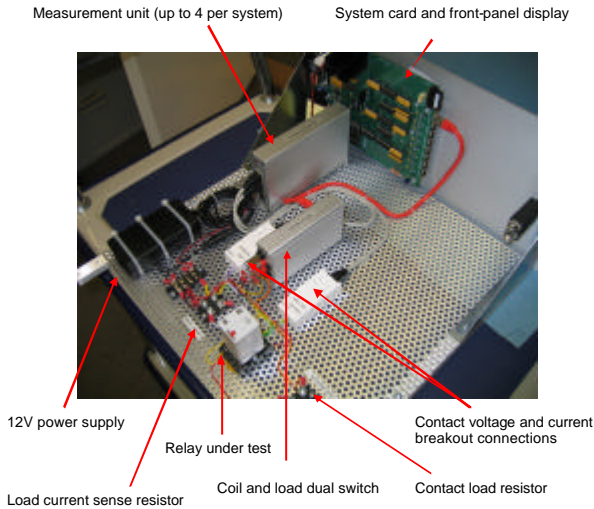


Figure 2 - The Reflex 51 Mark 1 'concept' vehicle

The idea behind this physical construction was that we would create a low-cost base platform that could contain various standard hardware modules – the measurement channels and switches – from which a customer could create his own test system, probably using his own existing relay device environment and loads. The construction was influenced by a desire to 19” rack mount a number of these Reflex 51 sub-systems, probably dedicating a single Reflex 51 sub-system to each device under test. As a result, loads of up to 1kW could be physically located alongside the device within the sub-system, making larger life-test systems easily scalable [3].

To go with this simple hardware concept we produced a completely fresh way of creating relay life tests within a test

programming environment that brings together the latest ‘event-driven’ programming concepts from visual language programming and from working with programmable logic controllers. The principle is shown in Figure 3 below:

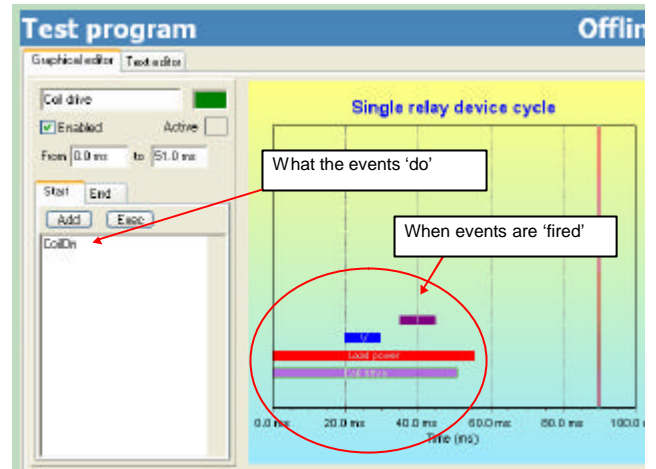


Figure 3 Test programming environment

The test program operates much like the pianola player piano with the punched music rolls. Every cycle of the life-test takes the system through one or more ‘time slots’ within the cycle. Each time slot has ‘start’ and ‘end’ event groups which cause the actual hardware actions such as to energise the relay coil or apply the load. This software ‘disconnection’ of the hardware action from the time at which the action occurs greatly simplifies the design of the test cycle program. Since these events can be hardware commands, measurements or expressions, simply programming the actions listed in the table below causes the relay to cycle and performs the life-test.

At time	Action to perform
0 ms	CoilOn
20 ms	VcontactStart
30 ms	VcontactStop
	CR:=Vcontact/1.2
50 ms	CoilOff

We showed this concept vehicle with its hardware and software to a number of potential customers and it was clear that this system could form the basis of an entirely new concept of testing, particularly since it could be used together with existing test loads and was highly cost-effective.

## II. THE REFLEX 51

Responding to customer comments we commenced development of:

- A general-purpose control system that would form the core of any implementation of this test system.
- A number of switch modules capable of a range of DC and AC switching.
- Measurement signal conditioning modules designed to interface between the device environment and the control system.
- Mechanical housing that would suit both rack-based and stand-alone implementations.

The control system was designed into a small card frame which in turn was designed to be mounted either on to a rack platform (as in Fig 4) or housed in a portable instrument case (as in Fig 5).



Figure 4 Rack implementation



Figure 5 The portable instrument

Providing this flexibility at the core of the system has allowed us to make REFLEX 51 acceptable for a very wide range of applications:

- Laboratory style experimental rig which can be easily re-configured for lab style life testing e.g. switch contact material testing.
- Large multi-chassis rack based systems with individual on-chassis loads e.g. automotive power relays.
- Large multi-chassis rack based systems with external loads. e.g. updating existing Life Testers using existing loads to 'bolt on' a new and improved measurement and reporting core.

RJ45 network cable connections are provided to plug easily and clearly between the control system and the device environment modules (Figure 6).



Figure 6 Easy connections from control system to the device environment.

It was considered vital to allow a user to quickly configure his own device environment by making connections between the Reflex51 control system and a variety of user custom wiring. This connection is made by means of local interface modules that isolate the control system inputs and outputs from the actual device voltages and currents (much as with a programmable logic controller). These interface modules are of two types - semiconductor switches to control the relay coil (and possibly the load) and signal conditioning modules that connect to measure points in the environment such as contact voltage drop or load current. These two types of module share a common mechanical design as shown in Fig 7 below.





Figure 7 Measurement input signal conditioning module

A module case measuring about 10x10x4 cm is fitted with threads to permit mounting both on a DIN standard rail bar and almost anywhere within the Reflex 51 rack platform in 'Lego' fashion. This allows fan-blown relay device contact load environments to be created locally within the rack platform up to around 1kW total dissipation. The design of the rack platform front panel also allows for these modules to be mounted to provide total front-panel device environment access as shown in Fig 8 (You can also see the rear of these modules in Figure 4).

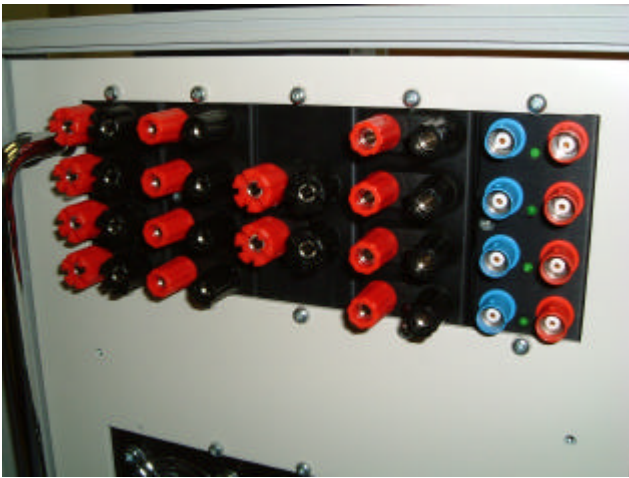


Figure 8 Environment modules mounted in the front panel.

The electrical flexibility with which different relay device contact load environments can be addressed is determined by the range of switch and signal conditioning modules that are available. A typical switch module is shown in Fig 9.



Figure 9 A typical semiconductor switch module

Many modules are quad switches, each module case housing four independent semiconductor switches. These are DC designs with switch ratings up to 30 amps or 400 volts per channel. Channels can be paralleled to achieve up to 120 amps. All switch designs are fully isolated allowing the output to float anywhere within the device environment so the switches can be mixed low or high side connected. For higher powers or other special situations, we have a switch 'exciter' module which is designed to drive solid-state relays for high-power AC operation or MOSFET power modules. An example of this is shown in Fig 10.



Figure 10 The switch exciter module driving a solid-state relay.

For measured inputs, all device environment signals pass through a signal conditioning module. These look identical to the switch modules but are fitted with isolated BNC connections as shown in Figure 7. All inputs are fully differential with a common-mode range of up to +/- 400V depending on module type. In-built module signal processing ranges from simple resistive series protection through passive division, RC filtering and active programmable gain options up to x1000. The benefit of applying the appropriate signal conditioning module to the device environment is that cable runs back to the measurement controller have little effect on actual

measurement performance since we are emulating the situation that exists with an active oscilloscope probe – a boon when constructing physically large test systems with high cable currents.

### III. BRINGING THE HARDWARE TO LIFE

Good hardware is little use without a flexible and intuitive software environment to program and control it. To achieve a multi-tasking program that would apparently be turning switches on and off and making measurements simultaneously we brought software concepts from the language of PLC programming and from the latest generation of graphical design languages such as Visual Basic and Delphi. At the core of the programming model is the concept of the device cycle shown below in Figure 11.

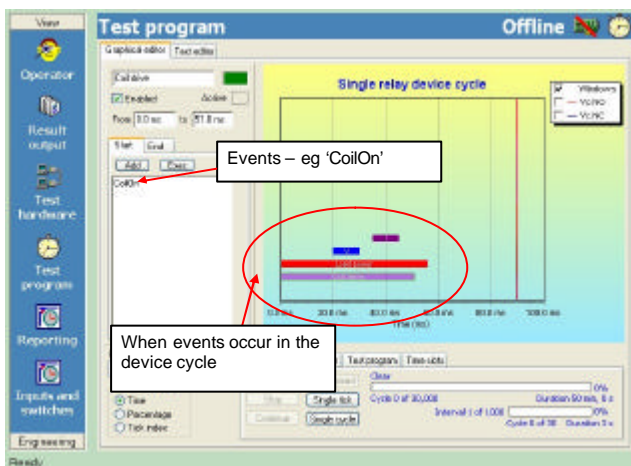


Figure 11 The Device cycle editor

On every cycle of a life-test, the Reflex 51 makes one complete pass of this device cycle. The user has already programmed the duration of the cycle (by programming the cycle rate). Within this cycle the user can program one or more time slots shown by the horizontal bars. Each of these bars is a placeholder for two event groups, one at the start of the bar and the other at the end. As an example, the event 'CoilOn' might be programmed to occur at the start of a timeslot and the event 'CoilOff' at the end. When the life-test runs, the action is to activate the CoilOn event at the time specified by the start of the bar and to activate the CoilOff event at the end of the bar. Dragging the bar through the cycle to reposition its timing moves the coil drive action accordingly without any numeric programming or changes to the event actions. By building up a cycle test program in this way the actions within the cycle and their relative proximity can be easily seen or modified.

You are probably wondering how you know what 'CoilOn' actually does in hardware terms. Figure 12 shows the command manager which provides access to the many

hardware and software events that the Reflex 51 can execute.

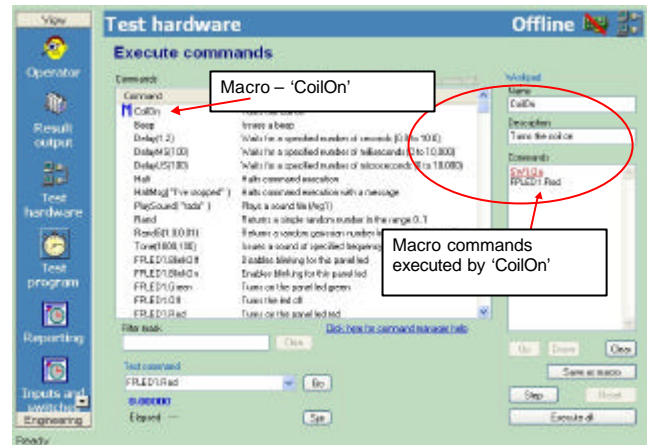


Figure 12 The command manager

The Reflex 51 actually implements a hardware 'language' designed to make the control of the switches and the making of measurements as easy and intuitive as possible, for example to activate a switch output you simply execute the command SW1.ON. The command SW1.OFF turns it off again. These simple switch commands could be entered directly into the event groups of a time slot in the cycle editor but they assume that you know intimately that the switch SW1 is (say) the coil drive switch. To allow you to forget the name of this switch and to provide flexibility to change hardware detail later, you can use a macro. Macros are collections of one or more commands named to appear as if they are an extension to the existing commands. Figure 12 shows this with the macro 'CoilOn' containing the actual commands:

- SW1.ON // turns the coil drive on
- FPLED1.RED // illuminates a front panel indicator.

In this case we are using a single, easily remembered macro command name ('CoilOn') to execute two actual hardware commands, a feature which dramatically improves the readability of the test program. The Reflex 51 provides the ability to execute any command or macro immediately in real-time, allowing you to test the command action or to debug your relay device environment without the complexity of the cycle timing. Being able to execute a command such as 'CoilOn' greatly reduces the possibility of error during hardware test or program modification.

Measurements are where the real power of the system appears however. Any time slot can be used to specify a range of measurement time over which a voltage will be averaged, so after a suitable settling interval for the relay device contacts to change state following the coil drive change we can instruct the Reflex 51 to measure a contact voltage drop. Measurements are made with reference to the

hardware input that is connected to that contact, for example the input 'M1'. To perform a measurement integration over a time slot we execute the command M1.START at the start of the time slot and M1.STOP at the end. From the stopped point onwards in time M1.VALUE returns the actual measured voltage averaged over the time slot and we will see in a moment how this gets turned into an actual life-test result. Since it is so easy to adjust the timing of the START and STOP points of a measurement you have full control over the measurement precision (longer times = less noisy values, shorter times = faster life-testing) and since the system can be instructed that the voltage is an AC voltage, you can choose exactly how many AC cycles over which to average the measurement.

All Reflex 51 events can be single commands, macros, or expressions that contain commands or macros. The expression capability enables us to produce (say) a contact resistance result from the measured contact voltage drop whilst allowing the test programmer to decide how this calculation is actually made. For example a simple life-test measurement requirement might be of a Form-A contact where we simply operate and release the relay and measure the closed contact voltage to derive the contact resistance – we assume that we know what the load current is expected to be and that this value is sufficiently accurate for our purposes. The test program events that we would program are:

Time	Event	Purpose
0 ms	SW1.On	Energises the relay coil
50 ms	M1.Start	Starts the contact voltage drop measurement
70 ms	M1.Stop CR:=M1.Value/10	Stops the contact voltage drop measurement. Calculate the contact resistance result from the measured contact voltage drop.
100 ms	SW1.Off	De-energises the relay coil

In this test program the cycle rate is 5 cps and the relay coil is energised for 100ms. During this time, the contact voltage drop measurement is performed starting at 50 ms and lasting for 20 ms (a good choice in a 50 Hz mains power environment to minimise stray signal pickup). The result of the measurement is obtained by the command 'M1.Value' which returns the actual contact voltage but you will notice that this function is used within the expression "CR:=M1.value/10" which derives a calculated contact resistance from this voltage drop and an assumed 10 amp load current. This contact resistance result 'CR' is calculated for every contact cycle and the action of making the 'CR:= ' assignment tells the Reflex 51 software that you want a test result called 'CR'. In this way you can have as many test result 'variables' as you like. Each test result is presented to you throughout the test in the form of a detailed line and distribution chart together with optional failure limits as shown in Figure 13.

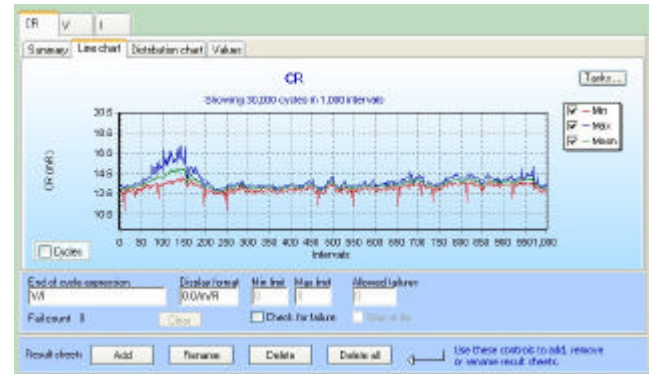


Figure 13 A typical result chart

Result charts can be customised to display in whatever format or units are required. When the test is run, the frequency with which the result is displayed on the chart and stored in the test program results is determined by the number of reporting 'intervals' that have been programmed. Reporting intervals allow you to compress huge numbers of contact cycle results down into meaningful max, min and mean values that retain extremes of values without having to manipulate all original measured cycles (although if your disk is large you can program single cycle intervals if you wish! – we've tested devices with a million intervals!).

Result data can be output either as simple exports to Microsoft Excel or in the form of a Microsoft Word compatible document which makes all measured data ready for viewing or printing and as shown in Figure 14 below.

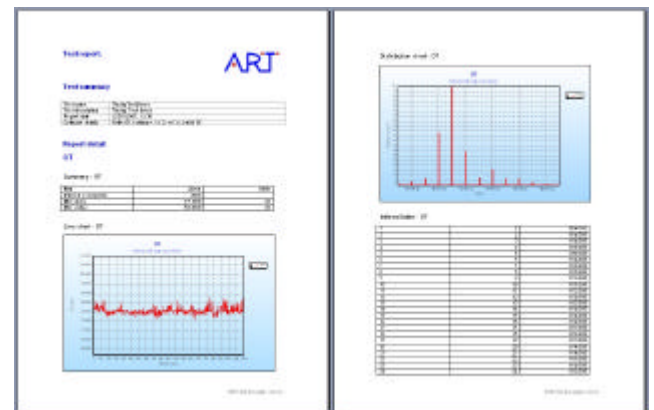


Figure 14 MSWord report output format

A more sophisticated (and precise) test program might measure load current in addition to measuring the contact voltage drop just described. If this load current was measured as a voltage drop across a 0.1 ohmsense resistor and using the 'M2' input, our load current could be a life-test result 'I' in its own right as:

$$I:=M2.value/0.1$$

We can generate a contact voltage result 'V' using the expression



$$V:=M1.value/10$$

And finally produce a 'CR' (contact resistance) result using the ohms-law expression:

$$CR:=V/I$$

The ability of the Reflex 51 to produce 'derived' results using expressions is highly useful since it permits you to make additional device tests at no cost or performance penalty from your existing measurements. Consider the case of a dual relay where it is known that the matching between the contacts of the two relays is an important measure of manufacturing quality. You might derive the contact resistance of each contact as shown above to produce the results 'CR1' and 'CR2' for each contact respectively and you would usually apply failure limits to each CR result in line with the allowed test limits ensuring that a 'good' part had all measured contact values within allowed limits. Further to this though you could create an additional 'quality inspector' result, for example called 'CRDiff' (for contact resistance difference) with the expression:

$$CRDiff:=CR2-CR1$$

which checks for matching between the contacts. Since this result can have its own failure criteria too, you now have an on-going quality test for the matching of the contacts without having to make any additional hardware measurements, perhaps weeding out parts which would otherwise pass all tests but have skewed individual test results.

#### IV. MEASURING CONTACT TIMING

We've just seen how voltage and currents are easily made within a relay device contact environment using programmed events and expressions. We realised however that being able to measure contact timing parameters from within that same environment without any additional hardware interfacing would be an extremely valuable aid to the relay engineer, particularly since it is likely that during a life-test run not only are the contact voltage and current parameters changing, but the overall relay device timing may also be changing due to mechanical wear or the onset of contact sticking that is as yet insufficient to cause a firm weld. Being able to assess this timing change by measuring it directly might be a great insight into actual device root cause of failure.

To this end we added the necessary system-wide triggering and measurement input edge capture hardware that now provides us with:

- Measured time to the first contact edge (operate or release time).

- Measured time between the first and last contact edge (bounce time)
- Counted number of edges (bounces).

It is important to note that these are actual measured values – not just go-no-go checks against limits. These values are as easily accessible on a measurement input as for contact voltages, for example the expression:

$$OT:=M1.TimeToFirstEdge$$

produces a result called 'OT' (operate time) which is shown in the result chart in Figure 15 below:

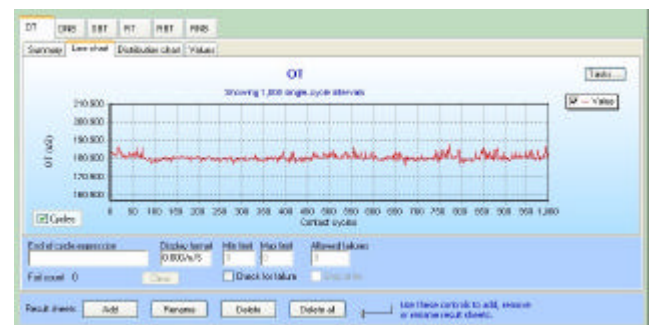


Figure 15 Measured operate time during life test

Of course together with this operate time result we also have other results available for release time, operate and release bounce time and numbers of bounces.....

The neat thing that we are really proud of is that these different types of results – measured voltages / currents, derived resistances, timing values – can all be combined into an output result group depending on your application and without major changes to the test program. Furthermore, if some values are measured only every few cycles – i.e. perhaps only every interval – you still get a life-test chart spanning the same test duration, it just does not have quite as many points on its chart. Very importantly though, whatever the result, it always has programmable failure criteria and so can still be used to halt the test on failure.

To extend this concept to its logical conclusion we are just completing an operate and release voltage module that will produce a measured operate and release voltage for a relay during the life-test in exactly the same way as just described. With this full parametric capability we will be able now to completely characterise a relay device during life-test with 'no stone unturned'.

Where next? This flexible relay device environment connection will be expanded to offer full turn-key parametric testing within the Reflex 51 software allowing for both single and life-test cycling modes of testing as well as to implement tests on other non-relay devices.

## V. CONCLUSION

This paper has introduced an entirely new concept in testing electronic components but using a low-cost module 'kit' of 'Lego' parts pre-configured for your common test requirements whether this be relay parametric testing, relay life-testing or the test of other non-relay parts (Figure 16).



Figure 16 - The Reflex 51 low-cost module 'kit'.

It has been shown that you can either configure a test system to your own requirements or simply enhance your existing equipment without the need to 'start over' with your device environment. A working demonstration has been made of this technique implementing a low-cost relay characterisation system that measures and records contact parameters and timing measurements routinely on every cycle during a life-test.

## VI. ACKNOWLEDGEMENT.

Applied Relay Testing Ltd wishes to thank its customers and its engineering staff for helpful advice and assistance in the preparation of this paper, particularly David Palmer, Stuart Hobday and Ian Flemming.

## VII. REFERENCES.

- [1] B.J.Frost "A new generation of test equipment" *Proc. of the 45<sup>th</sup> relay conference, NARM, April 1997*
- [2] S.J.Hobday "Bringing insight into the analysis of relay life-test failures" *Proc. of the 51<sup>st</sup> relay conference, NARM, April 2003*
- [3] B.J.Frost "Harnessing the power of the Web brings new possibilities to relay test equipment and data" *Proc. of the 52<sup>nd</sup> relay conference, NARM, April 2004*