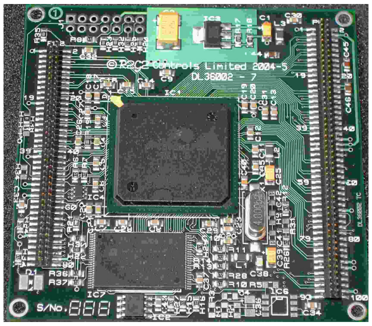
R2ZEN Introduction



R2ZEN is a complete generic processing core, consisting of a processor and a set of housekeeping peripherals, together with its suite of software. It is used in the R2G2 proprietary R2DCM rugged engine control unit, and the Optimiser dual-fuel controller.

The R2ZEN platform is designed for operating electro-mechanical systems, particularly at the systems development stage, such as engines, fuelling systems, transmissions and chassis systems where fast control system response is required and a wide range of input / output options need to be accommodated. The interfaces are flexible and adjustable to connect to a wide range of common sensor and actuator types.

For quantities of less than 100 units, the core is available as a plug-in daughter board, measuring only 75 x 75 mm. For larger production runs, we can design the core into the main board of a dedicated control unit to reduce manufacturing costs. Using this strategy, our customers have a controller perfectly matched to their needs – no unused features, no wasted board space. And if you need something unique to stay ahead of the game, we can build that in too.

Special Features

The R2ZEN architecture has been designed to meet the needs of customers with specialist applications and demanding requirements. The R2ZEN core concept arose from our customers' need for sophisticated controllers dedicated to their specifications – but at mass-production prices. This has led to a combination of features which are unique in this complex market place.

Fast Access to High Standards

R2G2 Controls Ltd has experience of both military and automotive philosophies. The former is renowned for rigour and reliability, while the latter brings responsiveness to customers needs and timescales. In recent years some military suppliers have been adopting the skills of the automotive industry to improve costs and time-to-market. At the same time, continuous improvement by automotive suppliers has led to reliability performance in excess of military specifications.

R2G2 brings sound practices from both arenas to meet the need for both short development times and consistent, reliable control systems.

Working in Real Time

For hard-real-time applications, it is vital that the controller is ready when the system needs it, and automotive systems can be very demanding.

Fast Start-up

In many real-time applications, the controller must be able to reach running state very quickly. In engine control, there is a need to support the start routine on demand and keep the engine running whenever possible. Momentary supply voltage blackouts must cause as little disturbance as possible. This leads to a need to start from a reset or power-up in a very short time.

The R2ZEN controller is able to initialise from a supply reaching a useable voltage to regaining control of a moving engine in less than 300 msec, while maintaining its high standard of safety checking. A running engine, for example, would thereby suffer only a momentary dip in speed if the power supply is interrupted.

Processing Speed

The processor is a 32-bit machine, running at 40 MHz, and uses a highly efficient instruction set. Furthermore, because the scheduling of tasks is strictly controlled, housekeeping and memory supervision are very efficient in their use of time. The result is an ability to run main computation tasks at a repetition rate of 10 msec, with measurement samples taken at 1 msec intervals or less. Thanks to the on-chip time processor unit, input and output pulse timing is to within 1 microsecond.

Security

The core contains a two-layer hardware watchdog system to ensure that no single failure can result in an unsafe condition.

Software Safety

Being targeted at vehicle systems from the outset, the hardware and software have been built around a culture of safety consciousness. The techniques required for high performance real-time control systems are often complex, so software testing and validation becomes complicated and vulnerable. R2G2 has worked for several years to refine a software design strategy which resolves this dilemma.

Complexity management

Complex software is fundamentally error-prone, due to the difficulty in detecting errors by review and testing, thereby complicating the validation task. R2G2 has developed a software architecture which significantly reduces the complexity of high performance real-time code and simplifies review and testing. R2G2's complexity management makes a significant contribution to software safety.

Expect the unexpected

Furthermore, R2G2 believes that it is not sufficient to prove that software is coded correctly, it is also necessary to expect the unexpected. It must be assumed that system components might fail, data might get corrupted or hardware might become physically damaged. The mechanisms for such failures may be predicted by FMEA analysis, but it is optimistic to expect an FMEA to predict all possible failure modes.

The R2G2 approach is to assume that all data is vulnerable, and to use a defensive, fault-tolerant software design methodology.

Choice of Programming Language

Different software languages have different strengths and weaknesses. R2G2 believes that elimination of bad coding practises, ambiguities and weak type-discipline make a more significant contribution to software reliability and safety than the choice of language alone. Reliance on a compiler having been validated is arguably naive. The R2G2 software design approach is compatible with implementation in either Ada or C, and the code is analysed and tested to the same high standard for either language. R2G2 normally uses a proprietary subset of the C language, heavily constrained to ensure safety and quality, and embodying many of the best concepts of Ada.

Safe Libraries

Many users of software languages rely on standard libraries and arithmetic operators, and cannot therefore know what problems they may contain. To avoid this, and provide total predictability, R2G2 code uses no bought-in library functions. Every line of our embedded software, including start-up code, maths functions, arithmetic operations, etc. are written and formally tested in-house, and provide protected results free from any risk of undesirable effects.

No Asynchronous Interrupts

In years of experience of writing and testing real-time embedded software, it has been found that a common source of unreliability results from interaction between code running at different interrupt levels. Techniques are available to minimise such errors, but the problem remains that testing exhaustively at all possible combinations of unscheduled interrupt timings is not realistic.

R2G2 has chosen to circumvent this problem by avoiding the use of unscheduled interrupts. Using a processor with time processing hardware, and taking great care in the design of the R2OS operating system, all incoming events are handled by polling. The only interrupt is the fixed timebase, from which all software paths are derived. R2OS also detects and logs any failure to complete tasks in the expected time slot, so that scheduling errors are handled gracefully and logged.

Data Protection

One type of failure which is becoming more significant in embedded equipment is random data corruption. The most significant sources of this fault are supply glitch data errors, EMC data upsets and RAM content disturbance by ionised particles. The degree of significance for each of these varies depending on the application. R2OS is uniquely designed to minimise the use of RAM for data storage. Code is executed directly from non-volatile memory and, along with all fixed data, is constantly re-validated by R2OS as a background task. All data stored in RAM is regularly refreshed and checked, including processor configuration data.

It is not possible to seamlessly repair all data faults, but techniques such as multiple stacks, stack checking, private data regions, write protection, runtime monitoring, multiple independent watchdogs and data validation provide a mechanism for early detection and logging of a wide variety of faults, followed by appropriate safe recovery.

Processor Exceptions

The R2G2 policy on exceptions is that they should not be used as a first-line defence against errors. Instead, defensive programming techniques handle errors and faults in an orderly fashion, logging their occurrence if appropriate. Exceptions, should they ever occur, are considered to be system crises, and are logged as such. Any exception event recorded by a controller is investigated by R2G2 to determine if there could be an underlying software weakness or a hardware fault.

Diagnostics

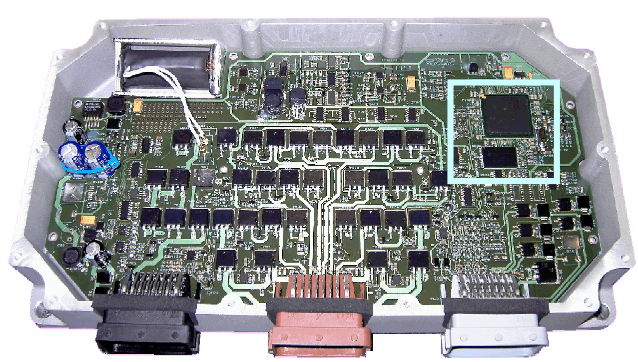
Each application contains full diagnostic analysis and reporting. Input data from sensors are validated to provide an assessment of the quality of the data. Output signals are read back to check that signals are operating. Where possible, inputs and outputs are correlated to estimate the 'sanity' of the surrounding system.

The results of these checks and many more are fed into a hierarchical diagnostic structure, which allows the user to detect the fault and interrogate the symptoms. Where possible the controller reports the source of the fault. The detailed data also allows the R2ZEN to handle faults gracefully and safely, entering reversionary modes or falling back to secondary sensors.

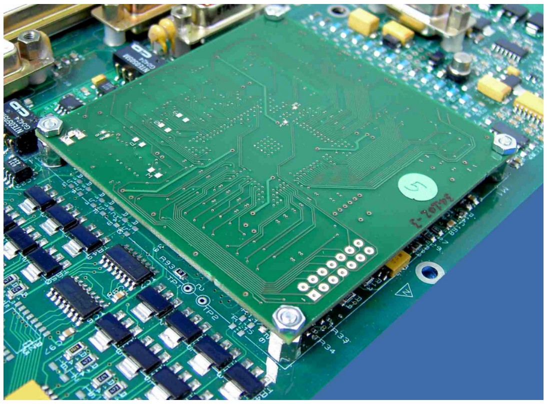
Adaptable, Flexible

From the outset, the design is intended to meet a wide range of needs and to be tailored with the minimum of effort. In the Optimiser single-board solution, the R2ZEN core is embedded alongside the switching and interface circuitry, yielding an efficient and cost-effective overall design for large quantity production.

In the R2BLU, R2MP, R2DCM and other controllers, the processing core is mounted onto the main board as a plug-in daughter module, giving flexibility and configurability. This is a simple method of creating prototype controllers, and leads to a lower cost of ownership during the R&D phase of a new product.



How R2ZEN is used by our Customers



Our customers usually get in touch with us before their project and specifications are fully defined. By selecting from the R2ZEN list of available interfaces, they are able to optimise the architecture of their system and streamline the design of their control unit.

R2G2 has a hardware library of interface circuits for the standard sensors and actuators commonly used in fuel control equipment, but we can also design new interfaces for novel sensors that our customers wish to use.

Blocks of interface circuitry are added to the control unit circuit as required, and the board for the unit is laid out to give a compact, dedicated product.

At the same time, the software drivers for each interface block are built into the application model, and coded into the executable package. New versions of software can be downloaded easily via R2Scope, and drivers are highly configurable to allow convenient integration of the unit into the vehicle.

How it Works

The following is a brief summary of the key specifications which give rise to this outstanding performance.

Hardware

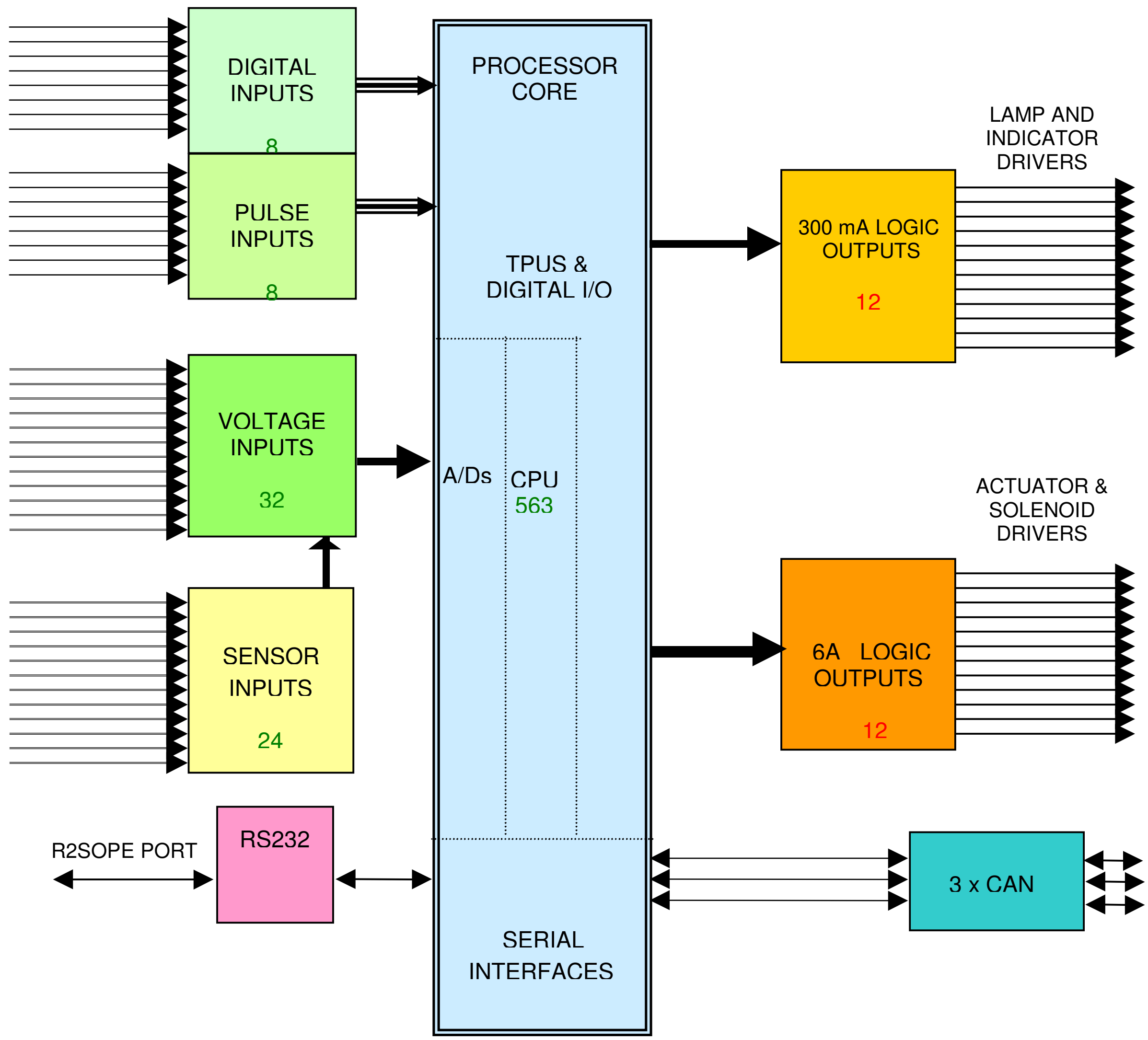
The structure of the R2ZEN core lends itself to a controller architecture with flexible blocks of interface circuitry around a standard centre.

List of Interface Channels

Processor Core Interface Type	Number	Example Use – Interface Blocks
Analogue Input	32	Pressure sensors Position sensors, LVDT, RVDT etc. Thermistors Gas sensors Pedal position sensor
Pulse Input / Output	32	Frequency and duty cycle of pulses Cam & crank pulse decoding PWM output sensors, pressure, position, etc. Injector driver output Actuator and lamp driver Serial data stream Tx or Rx
Logic Input / Output	10	Lamp / LED driving Actuator driving
PWM Output	8	Actuator / Positioner driving LED / Display driving Generation of analogue voltage outputs
RS232	2	R2G2 Scope diagnostic tool Inter-controller communications Diagnostics Serial sensors
CAN	3	Inter-controller Diagnostics

		CAN-based peripherals Dual-redundant CAN data-buses, e.g. MilCAN
SPI Interface	1	Interface expansion Serial memory device

Interface Block Diagram

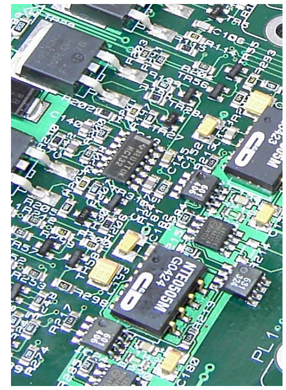


R2G2 Controller Block Diagram
Showing typical relationship between Processor Board and Interface Circuitry

Interface Flexibility

- Many of the interfaces are designed to be interchangeable; both Analogue and Pulse input types may be used as logic inputs.
- Analogue inputs may be configured to include pull-down or pull-up resistors, to permit open circuit and short circuit wiring fault detection.
- Thermocouple inputs are available as an option, and may operate with PRT sensors.
- When the Pulse Input / Output is configured as an additional serial data port, this may be used for streamed data from e.g. a GPS device or compass.

- One of the RS232 interfaces is normally used for the R2Scope calibration tool.
- The software includes facilities for testing, re-scaling and linearising measured values.
- Circuitry is available for operating with simple potentiometers, differential potentiometers, LVDTs, RVDTs and variable transformer sensor types.
- Several analogue inputs are selectable to monitor controller internal temperature, internal and external supply voltages.
- For extra-large and complex systems, the interface capability of the core can be expanded even further by multiplexing channels.
- The CAN ports are normally used to interface the controller with other distributed intelligence in the vehicle.
- Injector drivers are normally multiplexed, so that 6 cylinders may be operated by 5 output channels, 8 cylinders would require 6 channels.



The architecture of the hardware and software has been designed with safety in mind. Outputs are designed to revert to safe states, and the stand-alone watchdog operates in conjunction with the on-chip protection system.

The processing core is also equipped with the serial interfaces listed below, a stand-alone watchdog circuit and analogue-to-digital converters.

List of Communications Facilities

A range of serial data bus interfaces is available:
1) RS232 (for linking to portable computers), standard
2) Controller Area Network (CAN/J1939, used in automotive data networks), three channels, standard
3) MIL-STD-1553 (developed for military applications, using optional IC)
4) Two of the three CAN interfaces may be used as a Mil-CAN dual-redundant port, optional

Computing Resources

The central processor is a MPC563 device, running at 40 MHz. This is a 32-bit RISC processor, giving substantial computation power and accuracy, and the low clock speed ensures reliable cool running despite being in a hot environment. It is supported by 256 KB of Flash memory to store its operating system (R2OS) 192 KB of Flash for its application code, and 26 KB of RAM for storage of programme variables. Sets of tuning data may be edited temporarily or saved in the controller, streamlining the tuning process. The core also includes a 7MB flash memory for the log-book data storage. For a typical engine management application, the processor runs at about 15% loading.

Software

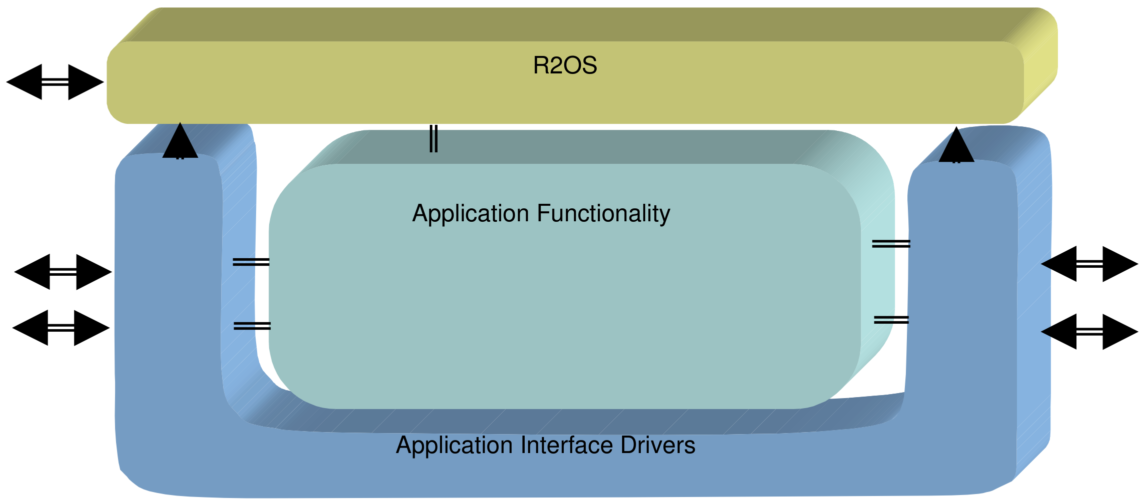
Although R2OS acts as an operating system for the purposes of the R2ZEN controller, it differs from the type of operating system normally found in desktop computers in a number of ways. The most significant of these arises from the need to avoid altering the operating system when far-reaching changes are made to the requirements. Since different applications often need to handle interfaces in very different ways, all interface drivers with the exception of the serial monitor port are in the application.

Thus the OS is responsible for the following tasks:

- Power-up sequence and hardware checks
- OS checksum and application consistency checks
- Reprogramming sequence and security
- Serial communications for monitor and editor
- Application task scheduling
- Scheduler crisis and data ownership handling
- Application serial data interface

The Application therefore contains the interface drivers as well as the 'functionality' of the controller. In many applications, the interface drivers are not altered and may be simply incorporated into the Application code, so they are treated as a set of accessible library functions.

The structure of the software may be represented like this:



The Application Functionality, which embodies the engineering relationships between inputs and outputs, is linked to the outside world via the Interface Drivers. All input signals are checked, scaled and linearised, while outputs are validated, so that the Functionality is able to operate in a protected environment with data of known quality.

The OS links to both regions, communicating mainly with the Application Functionality but also supervising the general health of the Interface Drivers.

All of the Application code may be readily re-programmed in a few minutes, allowing upgrades and comparisons during development.

Development Environment

In order to achieve the rapid development of applications that customers need, the support tools and infrastructure are vital.

R2G2 has developed a Windows-based interface tool known as R2Scope. This allows the user to monitor parameters numerically and graphically, to edit settings and maps, to download new programmes and to run automatic scripts of instructions. Many other functions and features have been incorporated to speed up the process of creating and tuning the control of systems.

Demonstration samples of this tool are available, and R2G2 will consider requests from customers for specific enhancements and dedicated features.

Physical Details – plug-in module

The processing platform is available as a PCB, 75 mm square and 7.5 mm deep, with four mounting holes at the corners. It is connected to its mother board by two dual-in-line connectors.

Example Applications

This section describes the general characteristics of typical application units which have been designed, based on the R2ZEN core.

Rugged Construction

The standard case is designed for aggressive environments including military use, and is sealed to withstand immersion or the low pressures of air transportation.

Environmental Performance

The case is designed to withstand operation mounted on or near the engine. The standard operating temperature range is from -25 to +85 °C, storage range is from -35 to +105 °C. The high-specification option gives an operating temperature range from -40 to +115 °C, storage range from -55 to +125 °C.

EMC

Many applications require high standards of Electro-Magnetic Compatibility. The R2ZEN processor core retains all high-speed databus signals within the processor to minimise emissions and susceptibility, and provision is made for interfaces to include appropriate RF filtering components. This is designed to maximise EMC performance.

The Origins of the Controller

The R2ZEN was originally designed to control a diesel engine by means of a common-rail or unit injector pumping arrangement. This task requires multiple high-speed outputs driving injectors with multiple pulses to a time resolution of at least 1 micro-second. This must be synchronised to cam and crank pulse trains, and controlled by sensors measured with a one millisecond sampling rate. Furthermore, the pulse decoder must handle safely any non-synchronised pulses on the input channels, to ensure safe operation.

Outside this, the engine speed control loop requires a choice of all-speed, two-speed and isochronous governing algorithms and dozens of sensors of various types. The R2ZEN processing platform was selected because it has the power and speed required for this challenging role.

A key aspect of the applications tackled has been the need to assist in the development of the mechanical system. This has led to special features being incorporated to facilitate logging results, analysing performance and adjusting parameters. Sets of data may be switched, uploaded and downloaded with ease, and optional routines selected manually. A built-in scripting facility enables the engineer to run a standard test repeatedly under a range of conditions.

The ancestry of the R2ZEN includes both military and industrial versions. This allows the range of performances and prices to suit a wide selection of customer types.