

## Customer story

# Hardware-in-the-Loop testing by Green Custard

## Raspberry Pi Pico supports powerful and robust automated testing

Britvic-owned Aqua Libra Co's sustainability plans depend on robust, reliable dispensing equipment. Green Custard designed a test solution that takes advantage of the flexibility of Raspberry Pi Pico to provide rigorous testing that would be difficult to achieve by other means.

Raspberry Pi solution	Raspberry Pi Pico
Size of business	SME
Industry	Testing, sensing and control

Under its "Healthy People, Healthy Planet" mission, Aqua Libra Co, owned by UK drinks giant Britvic, is working towards a sustainable future for workplace and retail drink dispensing. The company worked with Amazon Web Services (AWS) and Cambridge-based professional services company Green Custard to develop the Aqua Libra Flavour Tap, which eliminates the need for single-use packaging by delivering beverages straight to a reusable vessel. The tap can dispense still or sparkling water and a variety of flavours.

Within the tap are several control systems, one of which reacts to user input and relays flavour cartridge information to AWS. Another directly controls the dispense, switching valves and solenoids for precisely controlled periods as well as driving pumps for the exact quantity of flavour for a specific drink recipe. As part of Green Custard's work to help develop the tap, a solution for testing the software and hardware running on the dispense control board was needed.

The whole Raspberry Pi ecosystem provides a lot of information, which reduces development time significantly



## The challenge

The cost of finding bugs rises exponentially throughout the development and deployment process. Several studies have revealed the cost of resolving a bug, which can range from \$100 for an early change of requirements to many tens of thousands of dollars if fixed after the software is deployed. Reproducing an intermittent bug seen in the field can be very difficult, and unless the bug can be observed, it may well be marked as “unable to reproduce” and ignored.

Reviewing code can uncover implementation errors which could manifest as a bug, but testing is the cornerstone of modern software development. It takes many forms, including unit tests, integration tests, and so on. Testing software running on an embedded system can be particularly difficult, because often the unit under test requires inputs from external sensors which provide signals which change over time. A solution is to simulate these external signals, which allows the embedded software to function as if it were running within a complete system, not just an isolated test setup.

The inputs can be simulated using a technique called Hardware-in-the-Loop (HIL). With HIL, sensor inputs can be generated automatically, without user interaction. This helps to test how the software behaves under different conditions.

The Aqua Libra dispense controller consists of a microcontroller running a real-time operating system, and is used to monitor weight and fluid flow, as well as to drive stepper motors and electromechanical valves. Manual testing requires placing a vessel on a weight-measuring device to enable a dispense to be made. The subsequent valve switching and pump activity need to be monitored to confirm that the expected drink is dispensed. Along with this is the need to simulate the flow of liquid by generating a sequence of pulses like those produced by an electronic flow sensor.

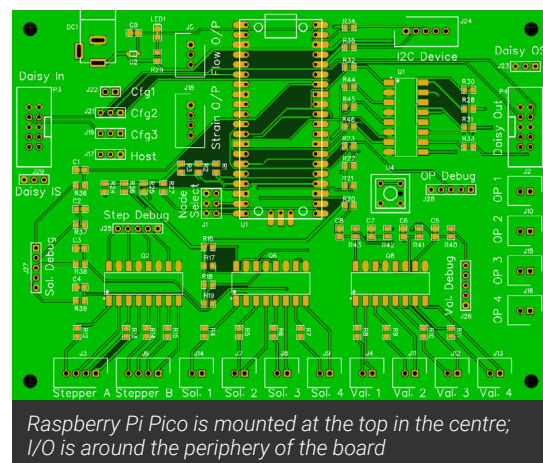
## The solution

Raspberry Pi Pico is based on Raspberry Pi’s RP2040 microcontroller, and has a peripheral which is very useful for HIL testing. In the test tool for Aqua Libra’s dispense controller, the Programmable Input/Output (PIO) block is used to generate very fast serial data, as if derived from a 24-bit ADC, to simulate a strain gauge. The ADC output can be set arbitrarily, and triggers events on the system under test. The PIO is also used to simulate the output from a flow sensor, and the simulated output is indistinguishable from that of the real hardware. The unit under test drives many stepper motors, and the PIO is used to read the

two-phase motor drive outputs, conditioned through optocouplers. This approach can measure every single step which would be applied to the motor, even when the step rate is in the kHz range. It provides unprecedented levels of instrumentation for a low-cost test tool.

Similarly, solenoid and valve states are monitored as the test progresses. RP2040 contains two PIO blocks, and a single stepper channel uses all the program for a single PIO. It is, therefore, necessary to daisy-chain multiple RP2040s to enable monitoring of all the I/O of the board under test. This requires a novel communication technique over serial that can address a unique board or all boards to configure for at the onset of a test, and recover the data as fast as possible during the test. The states of the I/O on the unit under test are sampled at 20Hz, and the collected data is written to a time series database together with a unique test identifier. Post-test, the collected data can be plotted using Grafana with a dashboard that is configured to represent a logic analyser. This is very useful for system development by the hardware and mechanical team.

The database can also be queried and, depending on the type of test, the data may be processed to signify a pass or fail condition. The whole system is fully automated and can run thousands of standalone tests without human interaction. The software running on RP2040 is identical for each test node, and the GPIO is used via jumper leads to provide a unique node identification which is read at runtime by the RP2040 application. This tailors the node functionality, loads the PIO programs for that node, and means that only a single image need be maintained.

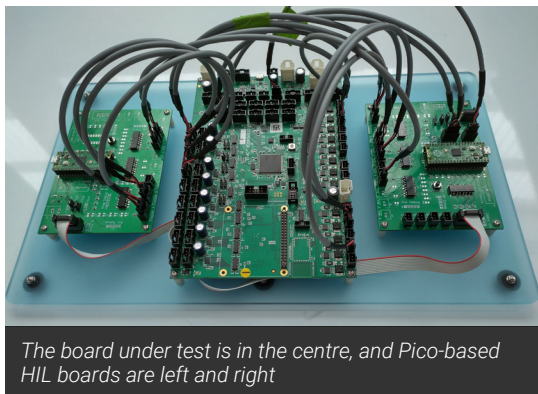


A PCB was designed and fabricated with a Raspberry Pi Pico on Node 0 acting as the main controller, with the option to connect multiple nodes to provide more input monitoring.

## Why Raspberry Pi?

Raspberry Pi Pico is attractive as a basis for this kind of testing for several reasons:

- **Cost:** the entire PCB with on-board power supply and flash storage can be purchased for less than £4. This is beneficial since it means several testing setups can be made for different sites, and if a Pico is damaged during development, the replacement cost is not prohibitive.
- **Form factor:** Pico uses 40-pin DIL headers, which can be soldered directly to a PCB or fitted with low-profile turned pin headers to allow for easy replacement.
- **PIO:** PIO, Green Custard reports, is “a superb idea, and gives such flexibility that it cannot be overstated how useful it is”.
- **Ecosystem:** The whole Raspberry Pi ecosystem – from blog posts to GitHub repos, YouTube tutorials, and the Raspberry Pi forums – provides a lot of useful information which reduces the development time significantly.
- **Availability:** Raspberry Pi Pico remained readily available for immediate shipping even during the global silicon shortage, so orders could be placed with confidence.



The board under test is in the centre, and Pico-based HIL boards are left and right

“Programmable I/O (PIO) is an incredibly flexible feature of Raspberry Pi Pico. It allows for the development of hardware interfaces that would be difficult without a full-blown FPGA implementation. The parts cost and the power provided by PIO made Pico a compelling choice. The SDK is well documented and there are many useful code examples in the accompanying GitHub repo. Pico is my preferred part for low-cost, low-volume development. I am pleased with the HIL testing framework. It gives me confidence that any change made to the software running on the target hardware has not introduced any regressions and performs as expected.”

– Neil Bryan, Principal Consultant, Green Custard

## The results

The flexibility provided by the HIL testing tool allows for many combinations of dispensed recipes to be run in an automated environment. A single test can request many thousands of dispenses which take hours to complete, checking each and every pump step and valve open-and-closure cycle. This would be almost impossible to accomplish manually. The test results can then be viewed either as text-based results, which are ultimately pass or fail conditions, or as a graphical display, similar to a logic analyser.

## The high level of detail and the ability to test many input conditions make for a very powerful testing tool

In Green Custard’s system, the test results are converted to records and written in bulk to an instance of an Influx time series database. The test input JSON file contains metrics which are used to determine whether the test has passed or failed. A Python script parses the JSON file, extracts the relevant metrics for that test, and stores them ready for use. When the test is complete, the Influx database is queried, and checks for specific features can be made; for example, the number of steps of the pump motors, or the valve opening time and duration to millisecond precision.

```
2023-05-16 17:00:48 INFO [TEST] starting: test 6 with tuid: 69e2744f
2023-05-16 17:01:46 INFO [TEST] writing 464 test records to the database
2023-05-16 17:01:46 INFO [TEST] tuid: 69e2744f start time: 2023-05-16T16:00:56.551063Z
2023-05-16 17:01:46 INFO [PASS] test case: 69e2744f start/stop conditions
2023-05-16 17:01:46 INFO [PASS] controller1: 1499
2023-05-16 17:01:46 INFO [PASS] controller2: 1081
2023-05-16 17:01:46 INFO [PASS] controller3: 501
2023-05-16 17:01:46 INFO [PASS] mdb weight total: 287
2023-05-16 17:01:46 INFO [PASS] htb weight total: 499
2023-05-16 17:01:46 INFO actuator1 opened at 0:00:00 for 0:00:03.372997
2023-05-16 17:01:46 INFO actuator2 opened at 0:00:04.412035 for 0:00:02.238067
2023-05-16 17:01:46 INFO actuator3 opened at 0:00:07.689258 for 0:00:01.119133
2023-05-16 17:01:46 INFO switch1 opened at 0:00:17.361552 for 0:00:12.704288
2023-05-16 17:01:46 INFO switch2 opened at 0:00:03.372997 for 0:00:01.639038
2023-05-16 17:01:46 INFO switch3 opened at 0:00:06.650102 for 0:00:01.039156
2023-05-16 17:01:46 INFO switch2 opened at 0:00:08.808391 for 0:00:02.638089
2023-05-16 17:01:46 INFO switch3 opened at 0:00:00 for 0:00:17.361552
2023-05-16 17:01:46 INFO [PASS] turbo transitions: 3
2023-05-16 17:01:46 INFO [PASS] test 69e2744f successful
```

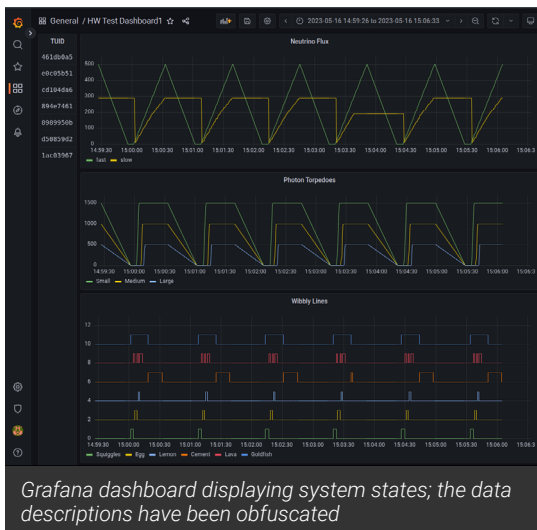
A successful test case

```
2023-05-16 16:59:34 INFO [TEST] starting: test 7 with tuid: cfb401ca
2023-05-16 17:00:40 INFO [TEST] writing 401 test records to the database
2023-05-16 17:00:40 INFO [TEST] tuid: cfb401ca start time: 2023-05-16T15:59:50
2023-05-16 17:00:40 INFO [PASS] test case: cfb401ca start/stop conditions
2023-05-16 17:00:40 INFO [PASS] controller1: 1499
2023-05-16 17:00:40 INFO [FAIL] controller2: 818
2023-05-16 17:00:40 INFO [FAIL] controller3: 0
2023-05-16 17:00:40 INFO [PASS] abc total: 287
2023-05-16 17:00:40 INFO [PASS] xyz total: 499
2023-05-16 17:00:40 INFO actuator1 opened at 0:00:00 for 0:00:03.613218
2023-05-16 17:00:40 INFO actuator2 opened at 0:00:04.412389 for 0:00:01.598092
2023-05-16 17:00:40 INFO switch2 opened at 0:00:03.613210 for 0:00:00.759179
2023-05-16 17:00:40 INFO switch3 opened at 0:00:00 for 0:00:06.050997
2023-05-16 17:00:40 INFO [FAIL] turbo transitions: 1
2023-05-16 17:00:40 INFO [FAIL] test cfb401ca failed
```

A test failure; controller1 is out of tolerance

## Using Hardware-in-the-Loop with Raspberry Pi Pico, Britvic can be confident Flavour Tap will perform as expected

A second, graphical, way of displaying data is provided by Grafana, an open source data analytics and visualisation platform. This view allows teams not involved in software development – for example, the people who develop flavours, and liquid specialists – to use the test data to determine whether the switching times and sequences are as expected. The image below, for example, shows three distinct charts that were generated sequentially during a dispense of four separate drinks. The top chart shows the data around the dispensing progress; the middle chart displays more data related to the dispense which changes over time; and the bottom chart shows the opening and closing of control elements within the tap.



*Grafana dashboard displaying system states; the data descriptions have been obfuscated*

This level of detail, together with the ability to test many input conditions, makes for a very powerful testing tool.

In this application, Raspberry Pi Pico enables Green Custard to provide a low-cost solution to simulate different types of sensors and measure the resulting change in outputs. It's possible to monitor the states of different components in real time, and write the collected data to a time-series database. Lastly, and crucially, the data can be plotted on a dashboard for further analysis. The whole process is automated and can run thousands of tests without human interaction.

Green Custard's Hardware-in-the-Loop system supports test cases that inject failure conditions that it would be very hard, or even entirely unachievable, to recreate on a real system, making it possible to assess the robustness of the software under test.

Without Raspberry Pi Pico, it would be very difficult – if not impossible – to identify intermittent bugs that could affect performance in the field. Using Hardware-in-the-Loop, Britvic can be confident that the Aqua Libra Co Flavour Tap will perform as expected.

To learn more about how Green Custard helped develop the Aqua Libra Co Flavour Tap, read their full case study [here](#).

**Connect with us**