

The Combechem MQTT LEGO Microscope

A grid enabled scientific apparatus demonstrator

J. M. Robinson^a, J. G. Frey^a, D. C. DeRoure^b, A. J. Stanford-Clark^c, A. D. Reynolds^c, B. V. Bedi^c,
D. Conway-Jones^c

Email Contact : j.m.robinson@soton.ac.uk

^aUniversity of Southampton, School of Chemistry, Highfield, Southampton, SO17 1BJ, United Kingdom

^bUniversity of Southampton, School of Electronics and Computer Science, Highfield, Southampton, SO17 1BJ, United Kingdom

^cIBM UK Laboratories, Hursley Park, Winchester, SO21 2JN, United Kingdom

Abstract

Grid computing impacts directly on the experimental scientific laboratory in the areas of monitoring and remote control of experiments, and the storage, processing and dissemination of the resulting data. We highlight some of the issues in extending the use of an MQ Telemetry Transport (MQTT) broker from facilitating the remote monitoring of an experiment and its environment to the remote control of an apparatus. To demonstrate these techniques, an Intel-Play QX3 microscope has been “grid-enabled” using a combination of software to control the microscope imaging, and sample handling hardware built from LEGO Mindstorms. The whole system is controlled remotely by passing messages using an IBM WebSphere Message Broker.

1 Background

As computer control becomes increasingly pervasive in the laboratory, it becomes easier to take advantage of automated and semi-automated experimental procedures to enhance the safety, throughput, and quality of data .

An absent experimenter’s expertise can be applied retrospectively and to future experiments but not immediately, unless they have the ability to monitor the experiment remotely (1). By adding remote control to this monitoring, the experimenter can then steer and modify the experiment without needing physical access. Whilst being of use to the local experimenter this technology opens the possibility of remote, realtime, collaboration with researchers and experts from beyond the local campus. In previous work the use of middleware to support a Publish-Subscribe (or Pub-Sub) methodology(2) for experiment monitoring has been investigated(3). MQTT(4) messages were passed by an IBM MicroBroker to relay information on the laboratory environment to remote users. The same technology can be used to pass messages from the remote user to the laboratory.

1.1 Safety and Physical Security

Remote monitoring and control technologies such as MQTT and the message broker can be used to implement “lights-out” operations of equipment. Though this can be problematic for laboratory personnel if the equipment or process is inherently dangerous. The use of local mechanical and electrical interlocks can help to avoid unintended local consequences of a remote action, and by adding remote monitoring the operator is placed in a more knowledgeable position. Security devices such as PIR sensors and door/pressure switches can be used to raise an alert if anyone unexpectedly enters a room or building. Identification technology such as RFID tagging

can also be used to verify the identity of personnel who enter or leave a controlled area, using the same MQTT message channel to alert the appropriate parties. Similarly local operators can monitor the status of remote controllers by similar means.

2 Implementation

2.1 Building the MQTT LEGO Microscope

The project can be broken down into three discrete sections, Physical Construction (the LEGO bricks), RCX Software (the link between MQTT and the LEGO sensors and Motors), and Microscope Software (controlling the camera hardware).

LEGO Construction

The LEGO construction around the microscope needs to support the microscope and sample, to provide sample selection/movement and focusing ability. The initial design involved focusing the sample by translating the microscope vertically. The weight of the microscope, and quantity of available LEGO dictated that it was more appropriate to move the sample vertically below the microscope, keeping the microscope static. The sample platform was driven via a rack and pinion lift powered by a geared down Mindstorms motor. A combination of gearing and driving the motor for very short time periods, allowed the fine control needed to bring the sample into focus even at the highest magnification settings. For protection the sample carriage was equipped with vertical limit switches. Both limits (top and bottom of movement) were connected to the same sensor input (wired in parallel) and the limit reached inferred from the current direction of travel. In a

further experiment the microscope was mounted on a LEGO tractor which could be driven over a floor mounted sample.

RCX Software

The LEGO RCX drive software is split into two parts. Both of these parts originate from IBM Hursley and were modified for this project. The LEGO IR tower and the IR receiver on the RCX are used to link the RCX to the MQTT network in real-time however due to the limited processing capabilities of the RCX a full MQTT protocol stack isn't used. The implementation used here uses single integers to trigger commands on the RCX and a two integer pair to receive a topic and message from the RCX for publishing over MQTT.

This solution requires two bits of code. One that runs on the RCX, and one running on a host PC, which bridges between the messages on IR link and MQTT messages over a TCP/IP network. To allow more extensive programming of the RCX, the standard LEGO firmware is removed and the "lejos" system installed instead(5). Lejos provides a basic Java based programming environment on the RCX and PC side libraries to enable communication with the RCX over the IR link. This makes programming interactions between MQTT and the RCX easier as the standard MQTT Java libraries can be used. The PC based bridge subscribes to a topic from an MQTT broker, and on receiving messages retransmits the payload over the IR link (where the message payload is assumed to only contain an integer). The software on the RCX receives messages over the IR link from the PC. If the integer received corresponds to one of the expected controls codes, the appropriate effector is triggered, otherwise the message is ignored. On receiving a sensor event (the lejos code allows event triggers to be tied to changes in sensor state) the RCX may perform simple processing on the data, triggering effectors as required, and then transmits a message over the IR link to the PC. This is shown schematically in Figure 1.

In the case of the LEGO Microscope there is presently one motor attached to motor control A which drives the focus position and two limit switches (one for each end of travel) connected in parallel to sensor input 1. The RCX accepts codes to stop the motor, and to run it forwards and backwards. It also accepts commands to "bump" the motor in either direction - this drives the motor for 50ms in the direction requested, then stops it. When the limit switches are triggered, the code firstly stops the motors, then backs the motor off in the opposite direction for 200 ms (to release the limit condition), it then publishes a message over the IR link stating which limit switch has been triggered.

Figure 2 shows this program flow schematically.

Microscope Software

The IntelPlay Microscope is viewed from the computer as if it were a webcam. It uses a cpia chipset(6), which is driven by the cpia.ko Linux kernel module. The module writer made control of the camera internal processing easy through the use of the Linux proc interface(7). Through this interface we are able to read and control Sample Lighting, and Image Brightness, Colour Saturation and Hue.

In the current system images are transmitted from the microscope using a simple webcam package(8). This presents images as either JPEG static frames, or as a multipart JPEG, moving them over a HTTP connection. It is planned to investigate other transmission methods, including sending the images over the MQTT broker, at a later stage. Control of the cameras internal processing and sample illuminators is provided by a Linux proc interface to the camera driver. This interface is presented as a file within the proc file system on the Linux machine controlling the microscope. Commands of the form "Toplight:on" are written into this file to control any of the parameters mentioned above. A Java MQTT client subscribes to a microscope control topic on the broker, and on receiving control messages writes the appropriate commands into the camera's control interface.

2.2 Frontend Control.

By using MQTT to pass messages within a remote control system frontend interfaces can be developed using a range of different programming languages, on a variety of platforms. For this demonstrator, the remote control interface was presented as a webpage. Within this page the user is presented with apparatus controls using HTML form elements alongside images from the microscopes camera.

The use of a webpage also provided the benefit that the web browser dealt with transfer and display of the images. The webcam server provided static JPEG images and a javascript timer can then be used to periodically refresh the image.

MQTT control messages are generated using a Perl CGI script running on the webserver. When the end user triggers one of the control buttons on the webform, the CGI script is called, which connects to the broker and publishes messages as expected by the RCX or microscope software.

3 Discussion

The use of LEGO components allows for the rapid development of mechanical solutions that may take weeks to produce using traditional manufacturing processes. The use of LEGO Mindstorms provides us with a convenient, robust, programmable controller which whilst limited compared to some of its industrial counterparts is adequate for this initial proof of concept. The use of LEGO components does bring

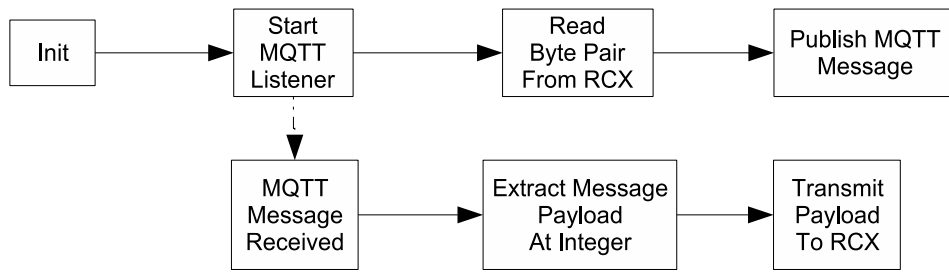


Figure 1: Software flow within the RCX PC Bridge Agent.

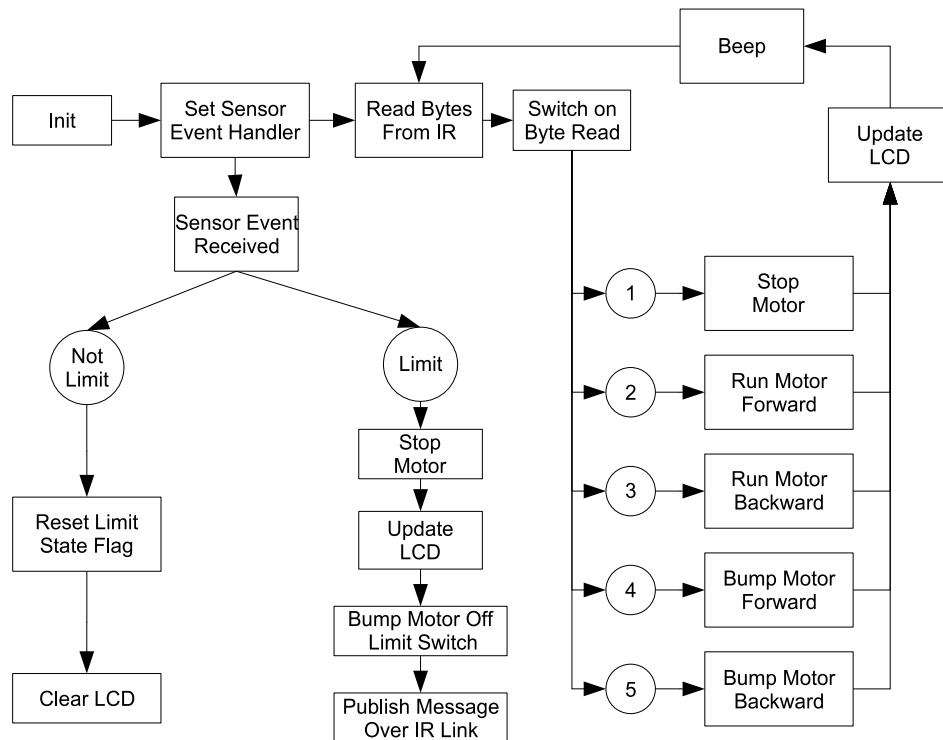


Figure 2: Procedure Flow within the RCX Code.

some limits in terms of repeatability and tolerance of construction. For longer term use high spec industrial components and indexed motion controllers would produce a system with a much higher level of reproducibility. LEGO also has a limited maximum load capacity (it is after all designed as a toy).

The use of MQTT as a remote control protocol brings some benefits and some unexpected features. By using existing libraries the software development can be extremely rapid, which when producing a demonstrator is important. MQTT also provides a solution to some of the traditional network problems, for example dealing with broken or unreliable connections, and making sure that the message always gets through. However, with remote control it is necessary to consider if messages should always

get through, if the end effector is unable to receive the control signal straight away, should the system then queue it for later delivery? It was found with this work that the reliability of the IR link between the PC and the RCX was somewhat variable, sometimes it was necessary to manually move the RCX to regain signal. When the link was regained, MQTT then ensured that waiting messages were delivered, often resulting in unpredictable behaviour. In the case of the LEGO model this was unexpected, but not dangerous. However, with another system the unpredictable behaviour from delayed messages could cause a dangerous situation, hence control hardware and software should be designed to avoid this, in particular using the non persistent messaging mode of MQTT.

4 Extensions and Future work

4.1 Sample Handling

As part of automating an apparatus in a laboratory moving physical samples onto and off the apparatus needs addressing. Some existing apparatus are fitted with auto samplers and sample batching handlers which allow the operator to preload a number of samples, usually into a grid or carousel that then sequentially presents the samples to the measurement position. By combining sample selection hardware with a scheduling and notification system the remote user could arrange a time on the apparatus during which their sample would be mounted, and they would be given control of the apparatus.

4.2 Inclusion of Remote Monitoring

Work on remote environment and apparatus monitoring has previously reported that MQTT can be used to monitor sensors in realtime(1; 3; 9). When implementing automation on a real-laboratory scenario, it is important to provide the operator with as

much information about the apparatus, and personnel in the vicinity of the apparatus as possible, both for reasons of safety and result reliability. By using common technologies adding this data becomes a far easier task.

4.3 Authorisation over MQTT

MQTT deliberately has a very minimalist approach to security, enabling appropriate security to be layered on top of it as required for any given application. In this demonstrator, security was purposely overlooked as the equipment is relatively safe, and data traffic limited to the campus network. Before using this type of system to allow control over a hostile network (such as the internet) an appropriate security scheme would need to be implemented. This could include; Encrypting the message payload (which could use PKI certificates to additionally provide signing for authentication and non-repudiation), Challenge/response security (by sending the challenge/response flows as MQTT messages over pub/sub), or TCP/IP traffic protection (such as standard VPN or SSH tunnels).

References

- [1] J. M. Robinson, J. G. Frey, A. J. Stanford-Clark, A. D. Reynolds, B. V. Bedi, Sensor networks and grid middleware for laboratory monitoring, in: First International Conference on e-Science and Grid Computing (e-Science'05), pp. 562 – 569, pages.
URL <http://doi.ieeecomputersociety.org/10.1109/E-SCIENCE.2005.73>
- [2] A. Stanford-Clark, Integrating monitoring and telemetry devices as part of enterprise information resources, Tech. rep., IBM (2002).
URL http://www-306.ibm.com/software/integration/mqfamily/integrator/telemetry/pdfs/telemetry_integration_ws.pdf
- [3] J. M. Robinson, J. G. Frey, A. J. Stanford-Clark, A. D. Reynolds, B. V. Bedi, Chemistry by mobile phone (or how to justify more time at the bar), in: Proceedings of the UK e-Science All Hands Meeting 2005, , pages.
URL <http://www.allhands.org.uk/2005/proceedings/papers/481.pdf>
- [4] IBM, WebSphere MQ Telemetry Transport Specification.
URL http://publib.boulder.ibm.com/infocenter/wbihelp/index.jsp?topic=/com.ibm.etools.mft.doc/ac10840_.htm
- [5] Lejos : Java for the rcx, accessed 24 April 2006.
URL <http://lejos.sourceforge.net/>
- [6] Cpia webcam driver for linux, accessed 24 April 2006.
URL <http://webcam.sourceforge.net/>
- [7] A. Nirendra, Exploring procfs, Linux Gazette 115.
URL <http://linuxgazette.net/115/nirendra.html>
- [8] jtravis@p00p.org, Official camserv home page, accessed 28 April 2006.
URL <http://cserv.sourceforge.net>
- [9] Floodnet: Pervasive computing in the environment, accessed 24 April 2006.
URL <http://envisense.org/floodnet/floodnet.htm>

IBM and WebSphere are trademarks of IBM Corporation in the United States, other countries or both.

Java is a trademark of Sun Microsystems Inc. in the United States, other countries or both.

LEGO and Mindstorms are trademarks of The LEGO Group in the United Kingdom, and other countries.

Intel is a trademark of Intel Corporation in the United States and other countries.