IOT SOLUTIONS IN THE FIELD OF THE MODEL RAILROADING INDUSTRY

Marco Ronchetti
DISI – Università di Trento
Via Sommarive 9, 38123 Povo di Trento, Italy

ABSTRACT

The model railroading industry offers products for the control of railway layouts, which are typically based on a centralized approach. IoT can be used for implementing instead a distributed approach, or a hybrid one. In this paper we discuss such alternatives, their advantages, and the possibility that future products could be based on such approach.

KEYWORDS

Model Railroading, IOT

1. INTRODUCTION

The first model trains origin at the end of 1800, when clockwork (spring-drive) and live steam locomotives were built as toys. They were not controllable, as once started they ran until out of power. After the first world war Frank Hornby started producing small-scale clockwork trains, which then evolved into electric models that allowed speed control by varying the current or voltage. It was the birth of a (small) industry that flourished reaching its peaks between 1960 and 1980, but which is still alive today: after absorbing many competitors in several countries, Hornby Ltd. (producing not only model trains) is one of the market leaders, and it is quoted at the London Stock Exchange. On the Tokyo Stock Exchange we find another industrial actor on the model train sector: the Japanese company Tomy. The market size is difficult to estimate, as not many data are available, but is not very big: over the last 10 years the annual sales of model trains in UK range between 10 and 20 million Pounds [Statista 2020], and in Italy between 7 and 15 million Euro [ISTAT 2020]. One of the biggest players, the historical German producer Märklin had a business volume of 112 million Euros in 2019. The turnover for Bachmann model trains (a Chinese manufacturer) for the year 2006 was approximately 47 million US$, growing to 147 million US$ in recent years, while the revenues for Hornby Ltd. in 2016 were 56 million Pounds. Other relevant manufacturers are the Austrian Modelleisenbahn Holding GmbH (53 million Euros in 2013), the Japanese Kato and the USA company Atlas Model Railroad. The reported business volume estimates for Märklin, Bachmann, Hornby and Modelleisenbahn Holding are from the German and English [Wikipedia 2020]. For the last two companies, as well as for Tomy, we do not have figures, but they should be more or less of the same magnitude as for Modelleisenbahn Holding GmbH. These figures from heterogeneous sources seem to be coherent and allow a rough estimate of the overall market size between 350 and 500 Million Euros.

Model trains are not really toys for kids any more: they present a high level of accuracy in the reproduction of the real trains, and are typically quite expensive. International standards exist: the European federation of national model railway associations (MOROP) publishes the collection of NEM-standards (Normen Europäischer Modellbahnen - Normes Européennes de Modélisme), officially in German and French [MOROP 2020], but unofficial translations in other languages are available. These standards are used by model railway industry and hobbyists in Europe. In North-America the standards are maintained since 1940 by the [NMRA 2020] (National Model Railroad Association). To some extent NMRA and NEM standards are compatible, even though in some areas they differ. These two standards are adopted also in other countries, like Japan, while Great Britain has traditionally used its own distinctive model railway scales which can rarely be found outside the British Isles.

Model railroading hobbyists often build train layouts at home: in most cases these present a limited level of complexity, due to space constraints. However, larger layouts are built in model railroad clubs, where enthusiast often gather: the oldest society is “The Model Railway Club” (https://www.themodelrailwayclub.org/, established 1910), located near Kings Cross, London, UK. In such cases the overall complexity can grow quite a bit. A famous case was the Tech Model Railroad Club (TMRC) at MIT, which in the 1950s pioneered automatic control of track-switching by using telephone relays.

Modular layouts are put together in special occasions like ad-hoc meeting or fairs. For instance, the “Friendship of European railway modellers” (FReundeskreis Europäischer MOdellbahner, FREMO) organizes meetings where scenery modules built and brought by individual participants are interconnected, reaching an overall length that in some occasions exceeded 1 km of miniature tracks.

In all these cases, monitoring, device actuation and train control is a (complex) need. Although classical solutions exist, IoT can play an innovative role in this scenario. In this paper we will briefly review the existing solutions and explore and suggest the advantages which IoT solutions can bring. These solutions could become products in the small but vivid industry of model railroading.

2. EXISTING CONTROL SOLUTIONS

Control in a model railroad layout is relative to various components:

- Train mobility (speed and direction adjustment, start and stop);
- Train routing (positioning of switching junctions, and operating turntables, transfer tables and railway level crossings);
- Signaling (by light or mechanical signals, depending on the “era” which is reproduced in the layout);
- Operating various kinds of accessories (such as e.g. lighting)

Solutions to the control problem depend on the type of technology being used to run the layout. At present there are essentially two options: analog or digital. We will briefly outline them here, giving only the necessary elements to proceed with our proposal. For a slightly more extended description of the two approaches, the reader could see [Ursu and Condruz 2017] and [Ursu et al. 2019].

2.1 Analog Power Feeding and Control

Since the advent of electric model trains, electricity was provided to the engines through the rails and the wheels. Each of the two rails carries one polarity. For this reason, all the railway configurations which bring to a train inversion, such as railway triangles (wyes), five-pointed stars and reversing loops, need to be managed to avoid short circuits. An exception is the German Märklin system, where the two rails carry the same current, and a central pickup shoe gets the other polarity from a “third rail” (actually a set of contact points in the middle of the tracks). Generally, train speed is controlled by changing the voltage on the rails. This modality is known as “analog control”. A significant disadvantage is that it is impossible to individually pilot multiple locomotives running on the same track, and hence they move simultaneously (even though their speed can be different, due to different motors and gears). To overcome the problem, the rails are electrically separated into subsection, and every subsection (or block) has a separated electrical feed and control. When a layout is divided into blocks in such way, every subsection hosts at most one convoy, trains can be individually stopped, and operators can run more than one train at the time without risking that a fast train catches and hits a slow train. This is typically based on sensors, which detect the position of the trains and allow reacting automatically in a prescribed way. Also, blocks can trigger signals or other accessories, adding realism. All this implies though a sensible complexity of the electrical wiring. Moreover, the routing problem (i.e. deciding paths for trains) implies deciding which current and speed controller should be associated to the interested electrical subsections. Solutions to the problem tend to be rather rigid, as they need to be hardwired in the cabling. Usually, the overall control ends up being centralized in a control unit where routing is operated, the right current is applied to the needed sections, and the speed of the various trains in different blocks is regulated.

2.2 Digital Power Feeding and Control

In the 1980s, the concept of electronic control was introduced. Initially the various producers defined their own systems, later a standard called DCC (Digital Command Control) was defined. The rails carry an alternate current, and play the role of a bus over which control signals are sent by using time modulation. Decoders installed on the locomotive convert these signals into direct current for the motor, so that every machine can be individually operated, even on the same track and without segmentation of the rails. The decoder also allowed adding new features, such as motion sounds, headlight lighting effects and cockpit lighting. Still, reversing loops and topologically equivalent configurations need to be managed to avoid short circuits. Also switching junctions, signaling and extra accessories can be operated by taking the signal from the rails, and using decoders placed near to the device that needs to be controlled.

However, apart a further increase of costs, the digital approach is not the ultimate solution of every problem. When there are several lines and several locomotives, it is necessary to install boosters that distribute the load, and cabling becomes necessary again. Troubleshooting in decoders, boosters or in the console itself can prove to be quite difficult. Moreover, digital control is a good solution when the “driver” sees the locomotive: for sections out of sight (e.g. shadow stations) automatic control is needed. A shadow station (or fiddle station) is a hidden part of the layout where trains are routed to tracks where they stop: in such way it is possible (also in a relatively small layout) to use a relative large number of different trains, so as to give the impression that a convoy has left to a remote destination, and activate it back later, hence alternating the active and hidden trains rather than seeing only a few, circulating in a loop over and over.

A solution to this problem can be found by using sensors and a computerized control which takes over and operates the individual trains and their routing. Dedicated software solutions of this sort are available and are able to control large installations such as e.g. the Miniatur Wunderland Hamburg [MWH 2020], which with its 1.300 m^2 large layout is the largest model railway in the world, and one of the most successful tourist attractions in Germany. However, once again the overall solution is quite complex, and inspired by the idea of a centralized control: the software running MWH is composed by more than 168.000 lines of code.

An alternative is to go back to the track segmentation idea used in the analog approach. For instance, the “hidden station problem” lends itself very well to such solution, and it is not the only case for which segmentation can be an interesting option also in the case of digital control. Of course, this reintroduces the (wiring) complexity that the digital approach promised to avoid.

3. PROPOSED IOT-BASED SOLUTION

3.1 Background

As we have seen, the type of control available for analogic layouts is highly centralized, with a control console where decisions are taken, mostly according to predefined scenarios which have been hardcoded in the wiring, or by acting on single actuators (such as e.g. the activation of single switching junction, while correspondently making sure – by hand – that the right current is given to the interested blocks).

In digital layouts we also have a central control unit, from which all commands are coded into the bus. Slave control units can permit to participants to take control of a locomotive, driving it, and possibly to operate switches and accessories in a pointwise manner (it is also possible to predefine sequences of switches, to be operated simultaneously). In general, in this case the control of the power source assigned to a given track does not come into play – since the current is the same everywhere. The issue of segmented blocks reappears though, as we discussed, in the case of non-linear, hidden portions of the layout: sensors can be used to cut off the power in a segment, so as to block a train. With respect to the analog approach, segmentation is slightly simpler since it is not necessary to distinguish which power source is connected to each block: on the other hand it is necessary to remember which train is where, since now their control is linked to their identity (their decoder ID) rather than to their position.

In our view, IoT offers the possibility to rethink the paradigm, taking an alternative approach, which we believe offers certain benefits that we will discuss. In this approach, every routing, signaling and accessory device becomes an IP entity. Also the speed control can become an IP entity for analog driving, as the power to a given block can be issued via Pulse-Width Modulation (commanded by a microcontroller), while in the case of digital driving, the DCC signal can be issued by a computer program (see e.g. the jmri project [JMRI 2020]), and hence it is manageable from the Internet.

3.2 Hardware Aspects

Implementation of control can be cheaply done thanks to the advent of microcontrollers like Arduino and its clones, and thanks to a wide variety of available sensors. In recent years all this boosted a lot of initiatives such as the Fab Labs, where mostly young students approach various types of automation problems. Whole new fields, such as e.g. Educational Robotics have emerged. Books, on-line tutorials, starting kits are flourishing everywhere.

Connectivity is an essential ingredient for implementing an IoT solution. Although Arduino add-ons allow extending its capabilities with Bluetooth or other types of wireless connectivity, we find particularly interesting microcontrollers like the ESP32: a series of low-cost, low-power system on a chip with integrated Wi-Fi and dual-mode Bluetooth. They were developed by Espressif Systems, a Shanghai-based Chinese company, and are manufactured by TSMC using their 40 nm process. What makes them particularly appealing is their connectivity capability, and the fact that they are based on a dual-core microprocessor (Tensilica Xtensa LX6). They include built-in antenna switches, RF balun, power amplifier, low-noise receive amplifier, filters, and power-management modules. Moreover, they can be programmed with the Arduino development kit, which is quite simple and for which there is a lot of documentation available. Hence with a few Euros we can get a relatively powerful microcontroller with a large number (over 20) GPIO pins (General Purpose Input/Output), ready to be loaded with a Web Server, or able to act as a MQTT (Message Queue Telemetry Transport) client [Locke 2010]. A client library providing support for MQTT is available on github (pubsubclient) [O'Leary 2020]. The MQTT broker role can be played e.g. by a Raspberry PI using the open source Mosquitto software [Light 2017].

3.3 Architecture

The main idea draws on the analogy to the object-oriented programming (OOP) philosophy: the world is modelled by "objects", each of which encapsulates data and functionalities, hiding the local complexity of state management, and exposes "methods" (i.e. functions) to perform operations, most of which are at high level, but could also, if desired/needed, expose lower level functionalities. If desired, also detail of the internal state can be exposed, similarly to what "getter methods" do in OOP.

By analogy, the layout can be decomposed into functional units, each of which is managed by one (or more, if complexity exceeds a certain threshold) microcontroller. Each functional unit exposes something similar to an API, which is reified by the dictionary of accepted MQTT messages.

When a train approaches a functional unit, it will be necessary to send a message requesting a permission, with the declaration of the needed services. For instance, if the unit has multiple entry/exit points, the request will declare the desired destination, and the microcontroller will check if the request can be satisfied immediately, checking and possibly updating the internal state of the unit. This means, it will decide the routing, verifying that the path is available (i.e. not occupied by another train or by conflicting requests), apply the right current to the sections which compose the path, and finally grant enter permission to the requestor – or let it wait until the requested conditions are satisfiable.

As an example, a module having a station with several tracks could expose:

- *high level* commands, by which an incoming train could be asking to obtain a track where to stop without any further specification;
- *intermediate level* commands, by which the train driver could ask which tracks are available (the state) and then request a specific one, leaving the microcontroller to decide the routing to reach the destination;
- *low level* commands, which enable deciding every detail of the path leading to a certain position.

Ideally, the low-level details should be left to the microcontroller, so as to simplify the overall management, but if desired, the option of intervening even at very low level could be granted.

Functional modules could also be grouped into higher level abstractions: a single-track line with some station which allow the crossing of convoys going in opposite directions could become a single logical unit, which exposes its available commands. These would be then decomposed into the set of commands to be issued to the individual functional models. In this way the operator could simply delegate the control of one or more trains to the microcontroller managing the higher-level abstraction.

Of course, one could envision classical "hold and wait" cases in which a deadlock situation occurs. We will come back to this point in section 4.

3.4 Control

Having subdivided the overall system into functional units which expose the possible actions, we can now examine the control needs and possibilities. One option is to keep the centralized model that we had in the old (digital and analog) approaches. A central console can be built, from which commands are sent to the "periphery", i.e. to the individual functional units in the form of MQTT messages. Since we defined functional units as monads that can expose high level functions (such as "enter from point A and exit from point C"), low level functions (such as "turn a given switching point") and states (such as "track 3 is busy"), we have the freedom to decide the granularity level we want to achieve in the central unit. Of course, this also allows us to have a computer playing the role of central unit and offering even higher-level commands such as "go from station X to station Y via the mountain route, stop there for a while, and then reach station Z" without bothering with the single details.

This is an interesting possibility, but does not depart from the original model (centralized control), even though it makes its realization simpler: switches, buttons and lights on the central console would all be connected with a local microcontroller, which wirelessly interacts with the remote microcontrollers situated in the functional units. A relevant difference however is that now this option does not exclude other choices, which can be taken in parallel. In fact, it is not based on predefined wiring, but is accomplished by sending MQTT messages: also other actors are free to send messages, and hence we can also have distributed control. For instance, we could have a series of interlinked web pages, one for each functional unit: each page could show the state (if needed/desired) and the possible actions. In such case, the central control unit could also disappear, with the control being completely distributed. Such option would be ideal for cases like the FREMO meetings we mentioned in Section 1. For instance, a possibility would be to have NFC beacons close to each functional unit: a mobile phone entering in contact with the NFC beacon could load the (software) control unit for that module: e.g. a web page, or an activity in an Android or iPhone app. In this way, a player could follow and drive its train through the maze of the large modular layout. Obviously other solutions such as QR-codes could be chosen for recalling the right control unit.

3.5 A Simple Example

As an example (with only high-level commands), let us consider the management of a simple, small hidden station, which is fully managed by a microcontroller. The station is composed by 3 tracks where trains can be stored $\{T_1, T_2, T_3\}$. Trains enter into the station only right to left. An incoming train stops at the entrance E if unauthorized (waiting for authorization), then passes through two switch points $\{S_1, S_2\}$ which select the access to one of the three tracks. Each track has a sensor: when the sensor detects a passing train, it cuts off the current, stopping the train. When a train is restarted, the two switch points $\{S_3, S_4\}$ leading to the station exit are operated in such a way, that the routing is correct for the exiting train.

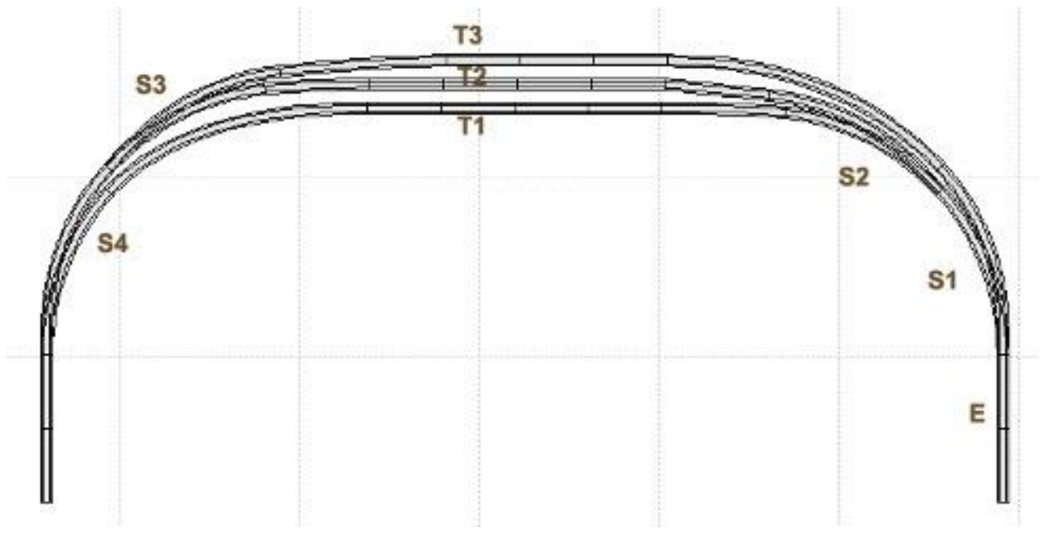

Figure 1. Our simple example: a 3-tracks hidden station

Overall, the station uses two actuators per switch point, and one – combined with a sensor - for every stopping position (one for every track Ti, plus the one at the entrance), for a total of 12 actuators. Using a traditional approach, each actuator would have a command switch, and the management of this simple station would require issuing several low-level commands. Using an IOT microcontroller instead, it's up to it to orchestrate the operations of the actuators. The microcontroller will simply expose two parametrized functions:

- Let train enter on a track Ti;
- Start train on track Ti;

The parameter i can lead to a deterministic behavior (if i has a value 1 to 3) or a non-deterministic behavior if i=0. In the last case, the microcontroller will arbitrarily choose a suitable track.

The MQTT message will be published on topic **`hiddenStation1`** with a Json payload of the type **`{"enter":"1"}`** or **`{"start":"0"}.`** As a response, the microcontroller will perform the needed checks and operations, an emit a message on the same topic with payload **`{"enter_effect":"i"}`** or **`{"start_effect":"i"}:`** a value of i between 1 and 3 will indicate that the successful operation affected track Ti, while the value 0 would indicate a failure (because the designated track was occupied on enter attempt, or empty on start attempt).

Additionally, we could foresee a payload **`{"status_query":"i"}`** which would generate a response message with payload **`{"status_response":"k"},`** where k=0 means track free and k=1 means track occupied.

Status-reporting messages would be interesting for instance for managing a central control unit with visual feedback, or as a means for a controlling software to elaborate the desired logic (probably concerning multiple functional units) according to the conditions (state) of the subsystems such as our hiddenstation1.

Of course, our simple example would easily scale to large similar station having N tracks: in the message exchange the only varying element would be the range of the parameter, which would run from 1 to N instead of from 1 to 3. The overall management complexity remains completely hidden, and the external interface would remain exactly the same.

4. DISCUSSION AND CONCLUSION

As we have seen, using IoT allows passing from the rigid, centralized control of an analog solution or from the essentially low-level centralized control of the digital solution to a more abstract, distributed and parallel control. One of the big advantages is the capability to expose high level functions, without the need to manually control every nitty-gritty low level detail.

Moreover, the untangling of the individual functional units thanks to the IP reachability, allows to more easily and flexibly compose distributed control units, which can be operated in parallel, while at the same time maintain the possibility of having a centralized overview.

This would make the control of a layout of medium or high complexity much easier and more flexible and would happily marry situations where the overall composition of modular component can be very large and informally structured, as in the FREMO meetings case. In similar situations, multiple agents could control in a distributed and parallel manner a multitude of trains over many modules.

As in all systems, where decision can be incrementally taken, there is the possibility of the formation of deadlocks, as we mentioned earlier. The classical conditions occur: *Mutual Exclusion* (no two trains on the same track), *Hold and Wait* (the train which arrives first occupies the track, waiting for the next segment to be free), *Circular Wait* (two or more trains are holding a resource and needing one held by another convoy) and *No Preemption* (trains cannot "disappear"). A typical strategy for the prevention of deadlocks is to avoid hold-and-wait situations. This is particularly important when most (or all) of the traffic is managed by a computerized solution. In this case, care has to be taken to prevent at least one of the four conditions: hold and wait can be the best candidate, as in the classical solution by [Djikstra 65]. When instead humans are involved in the control, we can rely on them and on their negotiations for the detection of deadlocks, and for their solutions, as in street traffic in the real world. And after all, in the worst case, in railway modelling preemption can always occur in the form of *Deus ex-machina* solutions (e.g. by physically removing a train by hand).

A more serious and complex issue is the fact that, as of now, the creation of a functional unit (such as e.g. the hidden station of our example) needs programming the microcontroller. Although programming an Arduino-like system is extremely simple, it is certainly not the type of thing that someone without a minimal specific education can do. Although this is true also for putting together tracks, which in the end are an electric circuit, the necessary notions and skills are in the case of programming more complex. For an IoT approach to have success also outside a small niche of skilled-enough people, it would be necessary to provide (almost) ready-made solutions, and hence to identify a series of patterns and to provide standard solutions for them, possibly with the need/possibility of minor adaptations.

Similar considerations hold for the building of the user interfaces provided via web or via smartphone app: also in this case it would be necessary to provide simple templates, which should to be easily customizable also for non-experts.

Hence, in spite of the many interesting advantages offered by the IoT solution, we believe that its applications are, as of now, limited to a small number of "railroaders". Some probably not too easy work remains to be done before such solution can be offered on the market as an out-of-the-shelf product.

REFERENCES

Dijkstra E.W. 1965. Co-operating Sequential Processes. *Programming Languages*. Academic Press, f. genyus edition, 1965

ISTAT 2020 - Data from http://dati.istat.it/Index.aspx?QueryId=8939, last visited November 4, 2020

JMRI 2020 Jmri project http://jmri.org, last visited November 4, 2020

Light, R. A. 2017 "Mosquitto: server and client implementation of the MQTT protocol. *The Journal of Open Source Software*, 2(13), May 2017.

Locke D. 2010 "MQ telemetry transport (MQTT) v3. 1 protocol specification," *IBM developerWorks Technical Library*, 2010, http://www.ibm.com/developerworks/webservices/library/wsmqtt/index.html.

MOROP 2020: https://www.morop.eu/index.php/de/nem-normen.html, last visited November 4, 2020

MWH 2020Miniatur Wunderland https://www.miniatur-wunderland.com/; last visited November 4, 2020

NMRA 2020 : https://www.nmra.org/index-nmra-standards-and-recommended-practices, last visited November 4, 2020

O'Leary N. 2020 "Pubsubclient, A client library for the Arduino Ethernet Shield that provides support for MQTT" https://github.com/knolleary/pubsubclient

Statista 2020 - Data from https://www.statista.com/statistics/377308/electric-model-trains-scale-model-construction-set-sales-manufactured-in-uk/, last visited November 4, 2020

Ursu, M.P and Condruz, D. A., 2017 Digital intelligent booster for DCC miniature train networks, *ModTech International Conference - Modern Technologies in Industrial Engineering*, 14–17 June 2017, Sibiu, Romania, IOP Conf. Series: Materials Science and Engineering 227 012133 (2017) doi:10.1088/1757-899X/227/1/012133

Ursu, M., Novac, O., Oproescu, M, Buidosó, G. and Hathazi F. I. 2019, Comparative Study of the Analog and Digital Operation for Miniature Railway Systems, 2019 *15th International Conference on Engineering of Modern Electric Systems (EMES)*, Oradea, Romania, 2019, pp. 137-140.

Wikipedia 2020, Data from the German and English Wikipedia, various voices: “Bachmann Industries”, “Hornby Railways”, “Modelleisenbahn Holding”, “Märklin” last visited November 4, 2020