A large scale CAN bus system.

By John Dammeyer
Automation Artisans Inc. Victoria, BC. Canada

This paper describes the design, manufacturing and installation of a CAN bus Lamp control system that eventually consisted of 765 nodes running a customized protocol updating nodes at 24Hz. A second network, with 6 nodes running the CANopen protocol, was used for diagnostics and bus configuration. Each LED Lamp node has a single processor that controls 36 Red, Green, Blue and White LEDs. The application software, written in Delphi and running on a PC sends messages via USB to a control computer containing 5 CAN bus channels. The 5 CAN bus channels were further broken into 3 sets of 50 nodes using a CAN Bridge. Additionally, with a USB to CAN controller the CAN open link into the Bridges was used to monitor the status of each of the 15 groups of 50 CAN nodes.What made this project unique was the time frame from concept to the design, construction and installation of this one of a kind product. Some of the problems, included rainy weather, component and cable failures are discussed.

The Project and the time line:

The project proposed was to create a set of LED Lighted 10m diameter Olympic Rings that would be mounted on the Lions Gate Bridge in Vancouver, BC, Canada.

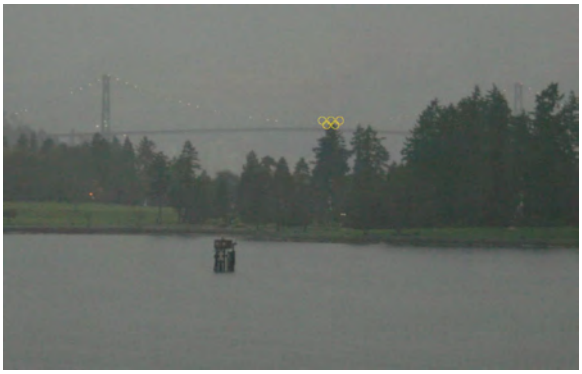


Photo 1 -- Simulation of Rings on Bridge "Lions Gate With Rings.jpg"

Power consumption was to be as low as possible while the coloured LEDs should still be visible from several kilometres away. The client also wanted colourful active light shows rather than a static set of white rings.

These rings were to be unveiled on February 5th, 2009, 1 year before the opening of the
2010 Winter Olympic Games. The first contact with the client was December 1st, 2008 where we were told the target installation date for the rings was the 20th of January, 2009. With Christmas and New Years in-between we had less than 8 weeks to deliver a working display.

By the 3rd of December, 2008 we had determined that other than wrapping the rings in metres and metres of COTS LED Christmas lights, the availability of any sort of commercial RGB (Red, Green, Blue) LED light assemblies was in the order to 8 to 12 weeks. We therefore had to build our own LED Lamps.

The Network:

Well documented and used everywhere for light shows is the ESTA DMX-512A specification for communication with theatre and decorative lighting. Running an RS485 bus at 250k bits per second with 512 slots for lamp intensity values, at first glance, it appeared to be the easiest solution. However DMX -512A is unidirectional.
We felt that since the venue was a bridge, service and access to the lights for potential firmware upgrades and status was considered critical. Therefore CAN bus seemed the logical choice.

I used Airy disk calculations to determine that from the unveiling view point we would need only 100 lights per ring to create the illusion of a contiguous circle. To reduce power consumption and prevent the rainbow effect of mixing Red, Green and Blue when displaying a white ring we decided to add White LEDs to the lamp fixtures.

CAN Devices have a 120 node bus drive capability and because the automotive market is full of CAN products there was no shortage of CAN based microprocessors. Eventually I found a Freescale 9S12 processor with 5 CAN channels. I ordered an Evaluation board from Italy.

Designing the Lights:

The first step was to fabricate a prototype LED panel with clusters of RGBW LEDs, resistors and a battery determine how many were needed and what sort of power consumption, intensity we could generate. I used a PIC18F CPU board from a different project and after a quick search on the Internet found an LED controller. The next day with the prototype parts on hand I built the first LED driver using narrow angle LEDs to get maximum on-axis penetration on foggy evenings.

Packaging and interfacing is the most difficult part of any product development. We decided to use an existing plastic Lamp Base with an O-ring sealed clear plastic lens. Experiments with a clear cap or a fresnel lens showed that the clear one created less prismatic diffraction.

After the initial experiments I settled on a panel of 36 LEDs in a 3x3 matrix, wired in groups of three LEDs per colour for a total of 12 channels. Control of intensity was through a TI LED driver and IIC bus. Access to the driver registers was also made available through a set of CAN messages so we could remotely check or modify the driver.

Ideally this sort of product would have one PC board that held LEDs, Processor Driver and voltage regulator. Due to parts availability and that we were using through hole LEDs we decided to break it into a CPU board and an LED board. This way assembly of the LED panels could start right away while the Surface Mount Technology CPU boards could be done very quickly later on. The LED panel prototypes were ordered and tested. Then a production order was placed and work was started on the software.

Meanwhile I designed the CPU board to use a mix of a switching regulator and a linear 5V regulator. With the arrival of the first 5 prototype CPU boards I assembled 5 lamps and was finally able to do some power consumption testing. With all LEDs turned on and polling CAN bus activity the power consumption averaged just over 4W at 24VDC which was what we expected. The panel of 5 was later used to test the PC show software.

Photo 2 – Lamp Test Assembly "TestBoard.jpg"

Connecting the lamps together became the next issue since we needed to provide power and network to 5 separate CAN systems.

Connecting the Lights:

By now the mechanical engineers had designed the structure to hold the lights and the electrical engineering department had drawn plans for the AC power and cabinetry. Placement was critical since we wanted to run the CAN bus at the full speed of 1Mbps. I could get away with 35m of cable at that bit rate.

We measured from the cabinet position to the last lamp and we were easily within the maximum distance based on driving each network from the centre out in both directions. This approach would put the 9S12 master node in the centre of the network with two

lengths out to the structure with 50 lamps on each end. Using this approach meant that current draw in each segment was half of the total for each network bus. Total bus length was also under 35m.

Again, after an extensive market search for CAN bus cables it became clear that the only cost effective solution (within budget) would be to purchase thick and thin DeviceNet cable designed specifically for CAN bus signals and we would have to build our own harnesses. Four pin female AMP CPC connectors were chosen to be on the thin cable stubs and we decided to connect the stubs to the thick cable using insulation displacement connectors.

The thick DeviceNet cable was slit, the shield spread apart and the wires made accessible. Although the displacement connectors were rated for our wire gauge the thickness of the insulation around the white and blue data wires made the connections difficult. Heat shrinkable boots were slid over the entire assembly and heat was applied to shrink the boots.

It wasn't until we started testing cables that we discovered that the technicians had been rather aggressive with the heat guns. A post mortem on a shorted cable showed the plastic around the displacement connectors completely melted along with the wire insulation. The shifting of the metal clips and wire fused the wires together into a solid mass and there were a lot of these.

The decision was made to strip insulation and solder the connections instead. That created it's own share of problems as quality control started to suffer with badly sealed connectors and unsoldered wires.

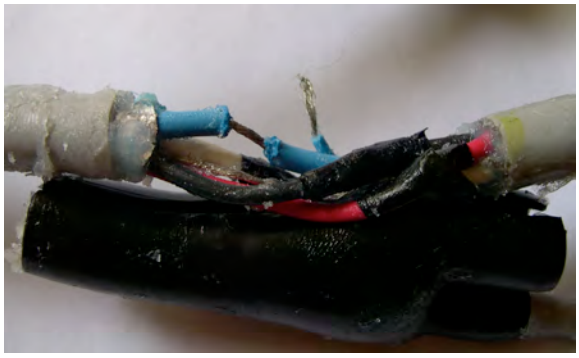


Photo 3 – Wire not Soldered "BadConnection.jpg"

By now more than a month had elapsed. A TDR was used to try and find the cable shorts and we were running out of time.

Power Supplies:

Cost was also a consideration when it came to choosing network power. We needed at least 425W per network but the thick cable restricted us to 7A which meant at a minimum the supply had to provide 14A at 32V. Ideally a 500W 36V supply would give us some headroom and keep us below the 42V maximum switching regulator input voltage and the bus voltage specification for the CAN bus driver. However, once again the deadline prevented acquiring anything in time and within budget.

The quick solution was to use a 24VAC 21A transformer along with a bridge rectifier and condenser. Unloaded the voltage sat at 38VDC and fully loaded it dropped to just under 30V. The electrical engineers also used a power line network analysis program to determine the voltage at the last lamp when all lamps were on drawing maximum current.. Too big a voltage drop and the common mode voltage between CAN signals and ground from the centre point driven end and the last node of the network would be exceeded.

To summarize the process for determining the power supply:

1. Node power consumption 4.05W
2. Number of nodes on bus section 50.
3. Cable maximum current rating 7A
4. Maximum switching power supply voltage on node 42V

5. Maximum voltage on CAN bus pins +/- 36V
6. Cable Length (17.5m) and resistance used to determine voltage drop along cable for common mode (-7V to +12V) calculations.

Control System:

Figure 1 now shows how the system was to be organized. From the PC for running light shows, through a USB connection to a 9S12 based evaluation board to 5 CAN bus channels. Each channel would drive a network of 100 devices from a centre point of a 35m long network thus distributing only half the power along each half of the network.

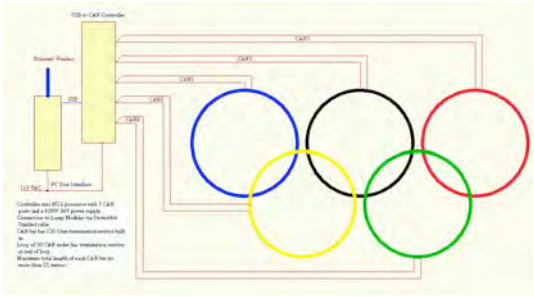
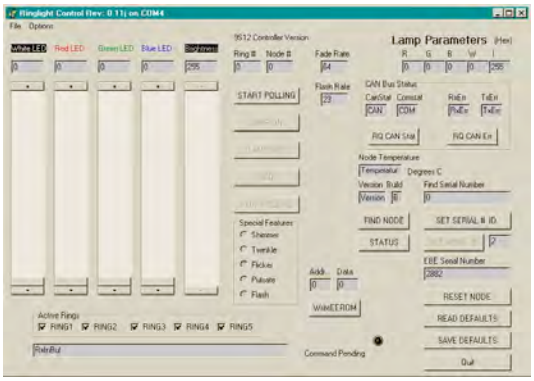


Figure 1 - Lions Gate Bridge Configuration "OrganizationLGB.jpg"

Before the 9S12 boards arrived I had already prototyped a small PIC based processor to act as a central controller. It had only one CAN port and sufficient memory for a 7 light data structure but it served as a test bench for the USB to Serial interface to the PC. This PIC became test module for polling and updating the nodes with lamp intensity information.

On the PC I wrote a test application that allowed me to examine registers and status information in any node; to even request the serial number from a node and to change the node ID using the serial # as a key.

From the start of the project we realized that replacing a lamp on the Bridge was difficult. We could easily have a node ID mix-up and so being able to change a node ID remotely, even if it currently had the same node # as another lamp on the network, was a fundamental requirement.



Screen 1 - Ringlight Control "RingLightControl0.02.jpg"

At this point I had a working PC application, controller software written in C for a PIC18F series processor and lamp software for the lamp PIC18F processors. I could turn on and off the RGBW lamps and adjust overall intensity and I could request status, including lamp temperature, from each lamp. I handed off the test application to a colleague who would use it as a template to write the show software to animate the lamps.

By now the 9S12 evaluation boards had arrived and I started porting the PIC18F controller template over to the 9S12. We settled on the Cine Camera frame rate of 24Hz for refreshing the lamps. This meant every 41.67mS we could conceivably supply a new set of lamp colours and intensities. If polling stopped for some reason, or the lamps lost communications they reverted to a default colour and intensity.

The Customized CAN Protocol:

The network now consisted of 101 nodes including the master and was meant to run at 1Mbps using 11 bit Identifiers. On average a message would be present on the bus for about 120uS but we wanted to design for worst case scenarios of 500kbps and 29 bit identifiers which meant an average of 135 bits or 270uS per message for a total time of 27mS. This fit into the polling window of 41.67mS.

To simplify things further we decided to pack the RGBWI information for two lamps into one 8 byte packet. Each intensity

value would now be only 64 levels with a group intensity
value of 8 bits. That's still 64x64x64 colours (262 thousand) each with 255 different intensities not including the white LEDs.

With the packed data we now had to transmit only 50 packets per channel every 42mS. That left lots of time for servicing the controller serial port. After each polling burst of 50 messages a request was sent out to a single lamp for status information. The lamp would respond, the 9S12 would then format that into an ASCI text message to send up to the PC.

The 9S12 evaluation board had a breadboard area that I populated with an FTDI USB245. This USB device could emulate a serial port from the PC side but had a byte parallel interface on the 9S12 side resulting in much faster transfers.

Using an ASCII protocol, we could debug with a serial terminal emulation program. Inside the 9S12 the messages were decoded into a 5x100 element data structure that was then packed into a CAN messages becoming even plus odd Lamp information and sent out every 41.67mS.

Venue Change:

Recall that Airy disk calculations show that a point of light expands and become larger as you move further away from the light. Correspondingly physical objects appear smaller as you move further away. Based on a progress report 4 weeks into the project, the client realized that the ring and lights structure would be roughly the size of a Canadian Penny held at arms length when the bridge was viewed from the unveiling location.

The entire project was placed on hold for over a week while alternatives were considered. It was now the middle of the first week in January and more than 5 weeks from the start of the project. We were ready to assemble 550 lamp panels. We had 550 CPU boards ready to assemble and final assembly of the enclosures had begun. About half the cables had been fabricated and tested.

The Vancouver Airport Authority (YVR) came to the rescue and offered up an ideal site for the rings and a location for the unveiling of the lamps. The problem was that this new location was now too close with only 100 lights per ring. They would no longer be rings but instead be circles of individual lights. The only solution was to add more lights placing them closer together. Calculations showed that we could just do it with 150 lights but that exceeded the drive capabilities of the CAN dri vers which were limited to 120 nodes.

After an extensive Internet search and help from members of the CAN List I found CAN bridges based on the same 9S12 processor we were using for the controller. Along with the ability to bridge between 4 CAN bus channels it also had a CANopen configuration and monitoring port. The supplier even ran our polling software on their evaluation board to insure their bridge would not lose any messages.
Now it was possible to drive 150 CAN based lamps with 50 per segment. And each segment was less than 35m long although several came very close.

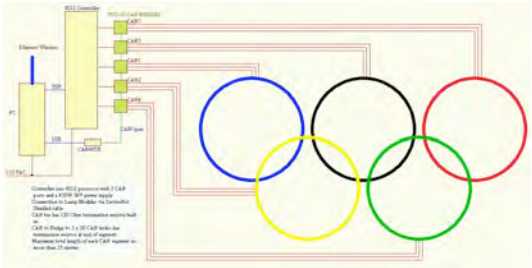


Figure 2 - YVR Rings with Bridges "OrganizationYVR.jpg"

The configuration became PC to USB to 9S12 controller to one of 5 CAN ports to a CAN Bridge to 3 CAN segments. Each segment still used half the current as specified in the original project. An added benefit was that the mechanical structures were built in sections for transport. It turned out each section held multiples of 50 lights so now a bridged ring segment could be identified discretely as for example Ring 3A or Ring 4C.

As luck would have it the transformers we had ordered were large enough to handle the 150 lamps. However, we now had to order an extra 250 LED panels and CPU boards plus parts along with 250 more enclosures and connectors and the labour to assemble
everything. We were given a one month extension. it was now the middle of January and the lights had to be on site by the 27th of February for installation onto the mechanical structure.

We would have two days to test the assembly while it was flat on the ground before it would be disassembled and moved to the airport. We would then have a few days to wire up the cables into the control box and test the system.

The Final Days:

The first on site days were spent wiring and adding connectors to the thick DeviceNet cable. Light wet snow falling with temperatures hovering around freezing made crimping the connectors difficult. Cable problems with short and open circuits and two lights responding to every single light message were just some of the problems. The most serious was finding 12VDC on the CAN bus. This turned out to be CAN bus wires inside the lamp touching sharp pins on the LED board.

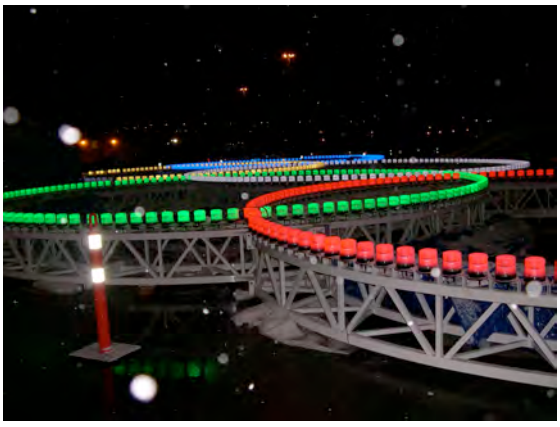


Photo 4 - Horizontal Rings - First Test "TestingRingsFlat.jpg"

With initial testing complete the rings were disassembled and taken to the airport site. Now the cables were cut to length and the pins crimped on and the segments tested.

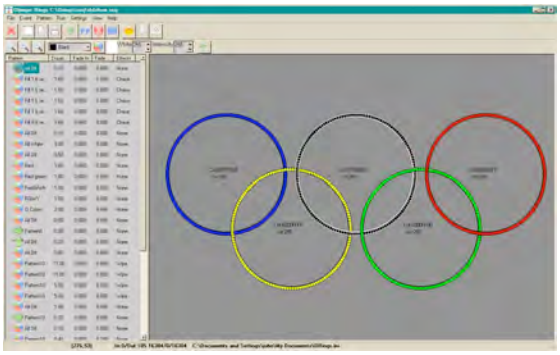
However, many of the lamps were installed in the wrong locations and therefore their ID numbers were not consecutive. That and several cables again had the power rail or ground shorted to the CAN bus wires. Some lamps again were bridging the internal 12V to the CAN bus.

Guy wires that held the structure stable prevented easy access to many of the nodes but working methodically we found the short circuits, defective lamps or connections and just before midnight the day before the unveiling the system was almost operational. Only a few lamps still needed their nodes set to the correct value.

Photo 5 - YVR Airport Rings "YVR_Rings.jpg"

Using my ring light program I entered in ID #78 and tapped the set node button. *And with that press of a button I instantly set 50 nodes to have the same node ID #78.* It was now past midnight. Fortunately the other two segments in that ring were powered down or there would have been 148 lights with the same #78 node ID.

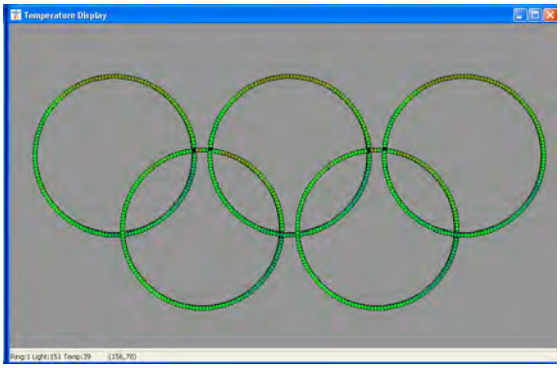
Originally, when we tested the lights and assigned initial locations we had recorded the serial # and the ID # of every lamp. To solve the problem we shut down power to all but the one segment. .For every node on the segment we had 5 possible serial numbers. An hour later we'd found each serial number and assigned the correct Node ID to the lamp with the matching serial number. The unveiling show went off without problems.



Screen 2 - ORingEdit Control Software "ORingEdit.jpg"

Operating for a year:

We ran into a number of issues over the next six months. Cable failures continued to cause segments to generate massive numbers of bus errors. I had neglected to consider the length of the bus inside the cabinet and two of the ring sets were so close to the network length maximum that we had multiple errors when the lights were cycling between off and full intensity; due to a combination of ground bounce and cable length.



Screen 3 - Node Temperature Display "temperatures.jpg"

Outside temperature variations would change a working segment into a failing segment due to expansion and contraction of the cable. On a record day in July the temperature inside some of the lamps reached 60C.

When we ordered the 5 bridges only 4 new ones were available.
To help us out the supplier graciously sent us their lab unit that had been subjected to EMI testing. Eventually we had to replace that unit as one of the three channels continuously produced errors.

The defective cables removed from the network generally had poor connections due to water damage. The defective lamps were always dead because of water ingress. One lamp had over 250ml of water and the level was high enough that the 36V supply caused the connectors and circuit board traces to be eaten away.

We finally made the decision to change the bit rate of the ring segments with the longest cables to 500kbps. We had to remove and reprogram the lights to do this. There were now some rings with dual speed CAN bus segments. Using the CANopen link into the bridges we modified the bridges to receive 500Kbps from the 9S12 and send either 1Mbps or 500kbps messages out to the ring segments.

Upgrading to Dual Rings:

Initially the dual rings were to be on land and would use the same hardware. We'd use two 9S12 evaluation boards, two USB ports and upgrade the software to deal with the two sides of the structure. However, a series of complications turned this project into another last minute race against the clock with specifications and changes occurring during development.

The passing of our Windows programmer due to complications from Cancer, lack of supply of the 9S12 evaluation boards and the need to run from a 48V battery system all caused unexpected problems.

The new lamp processor boards had two switching power supplies, one to convert the 48V to 60V battery rail voltage down to 12.6V and a second identical device to convert the 12.6V down to the logic 5V. When the prototypes arrived one was assembled, the code tested and production quantities ordered all in the same day.

To replace the no longer available 9S12 evaluation board and add relay control for power cycling I designed a small DIN rail 9S12 board with 6 relay drivers. We stayed with CAN Bridges and CANopen for diagnostics.

Photo 6 - New USB to 9S12 to 5x CAN "Assembled9S12-1.JPG"

This time the manufacture of the cables was contracted out to a firm with plastic over-moulding capabilities.
Even so, a shorted cable put 48V onto the CAN bus and instantly damaged 50 CAN drivers when power was applied. The drivers were rated for a much lower maximum voltage on the bus lines.

The end of the expensive supply cable was left as a coil on the metal deck before entering the distribution cabinet and This cable, carrying the 48V, 70A supply created an inductive induced over voltage spike that damaged over 300 lights when they went from fully on to fully off. Shortening the cable so it lay straight and flat along the metal hull of the barge removed the high voltage spikes.

Access to the system PC was via Remote Desktop and email. Our new windows programmer added email support so that we could send emails with a specially formatted subject and content to play a specific show.

Each day, at Midnight, the ring lamps were set to a colour of the day. The lamps would stay at that colour until one of the Canadian Athletes won a medal. The Project Leader [2] or I would then send an email representing the colour of the medal to a gmail.com email address. The Control PC checked for email once per minute and if it received a correctly formatted email it would run the show and leave the LEDs set to the colour of the medal won by the Athlete.

Photo 7 -- Double Rings in Coal Harbour [3] "Barge1a.jpg"

[1] Steve Corrigan: *Message priority inversion on a CAN bus* http://www.ti.com/lit/an/slyt325/slyt325.pdf

[2] Eddy Butler: EB Engineering Ltd. Sidney, BC. Canada

[3] Dave Nunuk – Photographer

John Dammeyer
Automation Artisans, Inc.
2335, Tanner Rd.
CA-Victoria BC V8Z 5P8
Phone: +1-250-544-4950
johnd@autoartisans.com
www.autoartisans.com