3000 RF Switches Networking
Low cost CAN based solution

Ran Weiss, LogiCAN ltd

There is a need to control some discrete outputs and monitor status inputs on each RF Switch module, in a network of up to 3000 such modules, using low-cost CAN based solution.

Overview

To control some 3000 low-cost units[1] (~$100), each located a few feet apart from the other, in an overall area similar to a concert hall.

The units are operated by a few ON/OFF signals and deliver their status via a few ON/OFF signals as well.

The whole system of about 3K units should be able to perform a user defined scenario (script) which is known before run time.

The requirements in detail were:

1. Centralized control of up to 3 discrete outputs, up to 3 discrete inputs on each unit.
2. Low-cost solution (< $6 per unit).
3. In a pre-defined scenario, all units should change their outputs in accuracy of 1mS. A scenario rate can range from 5/10Hz down to 0.01Hz (from 200/100mS up to 100S).
4. Drop-in installation (all units are identical, their location is the only difference) w/o particular addressing procedure.
5. The solution should preferably not employ software at the unit side.
6. PC based software only solution is highly preferable.

The Traditional Approach

A single copper-wire CAN bus may connect up to about 100 devices. This limitation is derived from hardware restrictions that evolve from CAN bus transceivers and cable capacitance and load.

[1] RF Sensor and/or Switch. But for this presentation purpose, consider it to be a simple I/O device.

Therefore, to connect 3000 devices, there is a need for about 30 separate CAN busses.

This solution is thus made of 30 CAN networks, each connecting up to 100 RF Switch devices. Each device should have an addressing setting feature, such as DIP switch or jumpers, in size of 7 bit (up to 128 selections). In an installed network, each device (RF Switch) must have a unique address.

In addition, a PC with 30 CAN interfaces is required. This may be implemented with 15 Dual-CAN PCI boards. In such a quantity, an industrial rack may be required.

The traditional solution is just no good. Looking at the 6 basic requirements, we see that most are just not met:

1. PC side using 30 CAN busses is expansive and the asociated software is more complicated.
2. Installation requires a unique address setting of each device (RF Switch).
3. The installation is more complicated, requiring separate 30 CAN busses.
4. Each device (RF Switch) requires a CPU+CAN chip, that though small, is running software that should be programmed into it, and be maintained. Therefore, device production is more expansive.

System is prone to installation errors and mistakes.

The Economic Approach

The approach is based on a “stand-alone” dual-CAN chip, the Infineon 82C900.

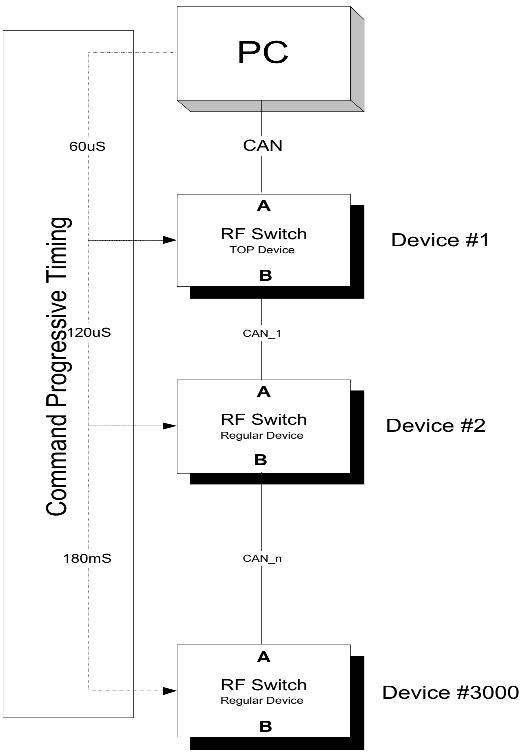
This chip supports the following:

1. Gateway functionality, between CAN_A to CAN_B. A message arriving at CAN_A, can be immediately forwarded to CAN_B, and vice-versa.

2. Message FIFO (between 2 to 32 messages). Such a FIFO can also be assigned to a gateway.
3. Stand-alone mode (w/o hosting CPU). In this mode, the 82C900 is self-initiated by reading serial EEPROM data, via it's SPI link. In stand-alone, the 82C900 offers up to 11 I/O pins, each can be independently be programmed as Input or Output.
4. After the stand-alone initialization, the 82C900 can be freely programmable over the CAN_A (or CAN_B).

Each device (RF Switch) housing one 82C900, offers two CAN interfaces: CAN_A and CAN_B. There will be only one CAN connection from the PC.

Each 82C900 will "see" only one more CAN node at each of it's CAN interfaces (CAN_A or CAN_B).



The devices will be connected in a "daisy-chain" manner, where:

1. device #(n) CAN-A will be connected to device #(n-1) CAN-B
2. device #(n) CAN-B will be connected to device #(n+1) CAN-A
3. Device #1 (the first in the link) will be connected to the PC with it's CAN-A connection.
4. Device #last (3000) will not connect it's CAN-B.

There will be only one "logical" CAN bus, that flows from one device to the next one. The devices are self- addressable, according to their location in the link. All control and programming are done from the PC, w/o the need to write and maintain software on the devices.

Due to the "daisy-chain" connection style, a message sent to device #n, will have to flow through all the devices located before it (#1 to #(n-1)). This will cause a constant arrival delay in the size of: n * message_time (message time is about 60uS for 1byte data @ 1Mbps).

To program a discrete output on an RF Switch, the PC should send one CAN data (data = 1byte) message on the CAN bus, with an appropriate CAN-IDy (y = 1 to 3000). At 1Mbps, this may take about (60*y)uS (60uS for device #1, 180mS for device #3000).

To read a device's (RF Switch) status, the PC should send one CAN remote (no data) message on the CAN bus, with an appropriate CAN-IDy (y = 3001 to 6000), and wait for a data reply message , on the CAN bus, with the same CAN-IDy. At 1Mbps, this may take about (150*y)uS (150uS for device #1, 450mS for device #3000).

Note that message rate is the same as in a standard architecture. This means that the PC can issue commands at the maximum rate of (1Mbps/60uS = 16,000 commands/sec). For example, to program an output on <u>all</u> devices, will take 180mS, the same as in a traditional approach of one CAN with 3000 devices.

Ran Weiss
LogiCAN ltd
51 Aliya-Shniya, Herzliya 46320 ISRAEL
Phone: +972.53.614188
Fax: +972.9.9597789
E-mail: ran_weis@netvision.net.il