Hi Guys,

In a typical alarm system scenario there are numerous **alarm panels** spread in a different armed premises.
Those premises are often spread in a very different geographical locations within a town or even country.
The best practice is to collect all the 'alarm events' from all **alarm panels** in a single **central station** .
So the question is what is the best communication media and protocol used for this.
Thanking into account the old infrastructure heritage it is not a surprise that the POTS analog telephone lines
have been widely used in the past and even now. And the protocol is **Contact ID** invented by a company called
**Ademco** which become de-facto a standard for this kind of **alarm panel <-> central station** communication.
It uses based on a sequence of **DTMF** tones with handshake and a control sum incorporated.
Here we go we have **AlarmSender()** Asterisk application implemented in **Switchfin**. (Note that **AlarmReceiver()**
supporting **Contact ID** protocol is available in Asterisk mainstream since Asterisk 1.2)

To test please build **Switchfin** with **AlarmSender** enabled in the **Switchfin** menuconfig.
Then in Asterisk **CLI** you can do

***switchfini1*CLI> originate DAHDI/g1/phone_number application AlarmSender* 111118162601000**

**-- Executing AlarmSender("DAHDI/phone_number", "111118162601000") in new stack**
**> AlarmSender: Setting read and write formats to u-law**
**> AlarmSender: Sending ademco contact ID**
**> AlarmSender: Determining protocol ademco contact ID...**
**> AlarmSender: Protocol ademco contact ID determined !**
**> AlarmSender: Trying to send for 1 time (max 4)**
**> AlarmSender : digits_to_send : 1111181626010006 with checksum : 6**
**> AlarmSender: Verifying 1400Hz 100ms burst (ACK)**
**> AlarmSender : Launching receving loop**
**> AlarmSender: Verifying 2300Hz 100ms burst (ACK)**
**> AlarmSender : Launching receving loop**
**AlarmSender: Sending DTMF digits successful !!**
**> AlarmSender : Launching receving loop**
**> AlarmSender : Not a Voice frame, trying with next frame**
**> AlarmSender: Ademco Contact ID sent successfully !**
**-- Executing [DAHDI@alarm:3] Hangup("DAHDI/phone_number", "") in new stack**

You should have **AlarmReceiver** (either commercial **alarm central** station or Asterisk box with **AlarmReceiver**
application)
listening on the phone_number side. The message is actually the last 15 digits.
For more information about the protocol itself please check Contact ID recommendation

The **Asterisk AlarmReceiver** application from the remote side responds:

***switchfini2*CLI>***
***-- Executing [6000@alarm:4] AlarmReceiver("DAHDI/Channel 1", "") in new stack***
***> AlarmReceiver: Answering channel***
***> AlarmReceiver: Waiting for connection to stabilize***
***> AlarmReceiver: Waiting for first event from panel...***
***> AlarmReceiver: Sending 1400Hz 100ms burst (ACK)***
***> AlarmReceiver: Waiting for first event from panel...***
***> AlarmReceiver: Sending 1400Hz 100ms burst (ACK)***
***> AlarmReceiver: Sending 2300Hz 100ms burst (ACK)***
***> AlarmMonitoring: Detected format ADEMCO_CONTACT_ID.***
***== AlarmReceiver: Received Event 1111181626010006***

This open the door for a few interesting applications based on IP0x as a component in the smart alarm systems.