Hard Air Gap - A device to isolate portions of a small network

An “Air Gap” in the networking world is a physical separation of network segments. Years ago I worked in a data centre and down the middle of the hall was a line of red tiles amidst the sea of grey. Under the suspended floor was a gap about 1m high where cables ran to the bottom of the cabinets that contained the servers. No cable was permitted to cross the red line - this simple rule enforced a physical separation between the potentially compromised areas of the network (REDNET) and the safe (GREENNET). The problem is, you still need to pass data securely between the two - consider online commerce; credit card and account data must be passed to secure servers in the rear from an internet connected network… there have been a number of very high profile hacks where account details have been stolen from non-air-gapped networks. HAG securely bridges these two zones - it is impossible to hack into the green network through HAG and the time gate limits the data exposure.

HAG supports 10/100Mb as only four of the conductors from the ethernet ports are connected through - gigabit support would require a doubling of the relays (but is do-able). Most things auto-negotiate this so you shouldn't be aware of it other than transfer rates - two minutes of 100Mb should allow you to transfer at least 40MB. If this is not enough, select one of the longer gate times, although any good transfer client (free or corporate) should permit resume and allow you to pickup where you left off when the NAS comes back, the client will just keep bashing away until the task is complete.

Air Gaps are now becoming more virtual (to match the change in server farms) and can be a combination of routing tables, network masks and firewalls. These provide the separation between networks but cost thousands.

This device (a HARD air gap - i.e. physical) was designed to switch two network segments to a single point based on a time “gate” in a small network and was in answer to a specific need.

The HAG needs to be supported by the networks on either side e.g. servers acting as go-betweens to marshal the data. All three servers need a NIC in the same network so they can see each other. Probably best to set up a small transfer network of 8 addresses and share it between these machines. The servers can have more than one NIC so it isn't a problem connecting them to their home network and the switched network. So for example

Server A has addresses 192.168.0.3/24 and 192.168.12.2/29 Server B has addresses 10.1.0.216/16 and 192.168.12.3/29 The NAS has address 192.168.12.4/29

Here, 192.168.0.x and 10.1.x.x are the home LANs for the respective servers and 192.168.12.x is the transfer LAN. Both servers can see their LAN and the NAS (when switched through). Each server pings the NAS, if there is no response, repeat until there is, then transfer any data.

HAG is not a network device in itself - it doesn't connect to the LAN but merely provides a `metallic` path from either of two Ethernet connections to one other. Think of it as a SPDT switch for 100Mb Ethernet. The common port should be connected to some NAS or other storage device (FTP server) and either of the other two connections go to the separated LANs. Each is connected for a time period in turn and the connections on the DIL relays mean they cannot be connected simultaneously. If both are on or both off, the network is effectively scrambled, making the device fail safe (i.e. both LANs are disconnected from the common.

The solution using this device is that server A pings the NAS. When it (begins to) responds, it has two minutes (or whatever your settings are) to move it's data to the NAS. Server B will later perform the same action to move the data off the NAS and/or pass back data to server A.

This solution is fairly easily scripted/programmed and if you use FTP (and others) you can resume your transfers when the network comes back round your way. Land the files with a non-exciting name and rename them after transfer so the other side doesn't attempt to process an incomplete file.

No direct network path exists between server A and server B. If your internet facing zone gets hacked, the attacker absolutely cannot jump through the HAG (or the common NAS) to your secure stuff on the other side.

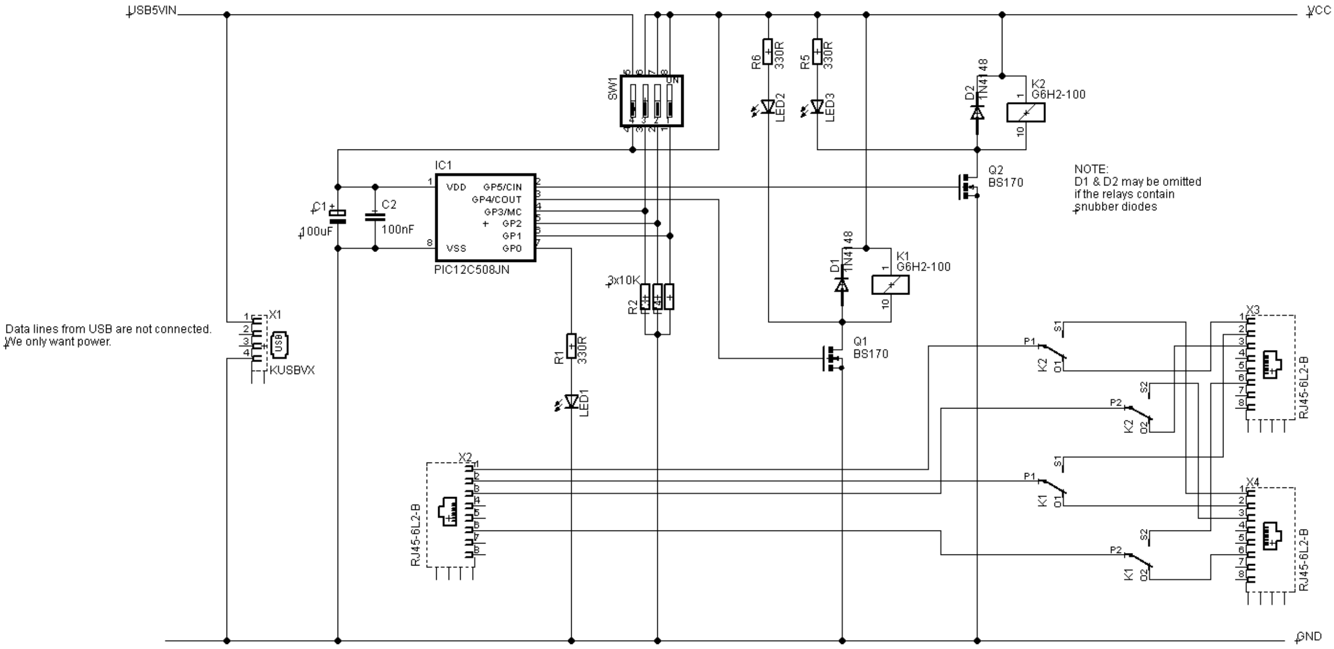
Finally found a use for a few dozen 12F pics. The below GCB code compiles into PIC assembler. The schematic is at the bottom of the page.

Future improvement would be to add a RTC so proper scheduling of access could be added. This would trigger a move away from the tiny PIC at the heart of this useful device.

HAG.GCB

```
' Hard Air Gap (HAG) Lite. A fairly dumb Ethernet port selector switch
' V1.0  20/11/2016  A.Henderson
'
' Controller code uses the tiny PIC12F508
'
' The code is not elegant but we are constrained by the PIC...
' also, this is all it does so there is no need to be "clever"
'
' Connects two ethernet PORTs (A, B) to a common port alternately.
' A relay (metallic path) matrix is such that if both are de/activated
' the wiring of the ethernet cable is scrambled thus it is fail safe.
' Relays must be activated alternately to connect correctly to the
' common port.
'
' The controller draws power at +5V, 50mA from a USB socket.
' No connection is made to the USB data wires
' No connection is made to the ethernet wires
'
' At startup, the PIC reads 3 dip switches on GPIO.1,2,3 :
' 000 = Off, both PORTs disconnected
' 001 = PORTA connected
' 010 = PORTB connected
' 011 = Toggle at 2 minute intervals
' 100 = Toggle at 10 minute intervals
' 101 = Toggle at 30 minute intervals
' 110 = Toggle at 60 minute intervals
' 111 = Toggle at 120 minute intervals
'
' Config is only read at power-up.
'
' There is no RTC so all timings are approximate
'
'***********************************************************
```

```
' ----- Configuration
  #chip 12F508,4
  #config osc = int, wdt=on
  #define Approx1s 996
  #define ConA  16
  #define ConB  32

START:

  ' assign the prescaler to the WDT
  'ASM [
  movlw   B'10001111'
  option
  ']
  DIM mm AS WORD
  DIM nn AS BYTE
  DIM RLMAP(2) AS BYTE '  this is the bit pattern to toggle
  DIM DLY AS WORD

  DIR GPIO b'11001110'

  ' GPIO.0 OUT    ' "Running" indicator
  ' GPIO.1 IN     ' dip switch bit 0
  ' GPIO.2 IN     ' dip switch bit 1
  ' GPIO.3 IN     ' dip switch bit 2
  ' GPIO.4 OUT    ' RL1 - PORTA
  ' GPIO.5 OUT    ' RL2 - PORTB

  Wait1s              ' pause a little after boot

  nn=(GPIO / 2) AND 7 ' read the switches (have to right shift)

  RLMAP(0)=ConA ' preset the most common config to save code
  RLMAP(1)=ConB
  DLY=10

  SELECT CASE nn
    CASE =0         ' disconnect both PORTs
      RLMAP(0)=0:RLMAP(1)=0
    CASE =1         ' PORTA static
      RLMAP(1)=ConA
    CASE =2         ' PORTB static
      RLMAP(0)=ConB
    CASE =3         ' 2 mins toggle
      DLY=120
    CASE =4         ' 10 mins toggle
      DLY=600
    CASE =5         ' 30 mins toggle
      DLY=1800
    CASE =6         ' 60 mins toggle
      DLY=3600
    CASE =7          ' 120 mins toggle
      DLY=7200
  END SELECT

Main:
  FOR nn=0 to 1
    FOR mm=1 TO DLY
      GPIO=RLMAP(nn) + (mm and 1)
      Wait1s
    NEXT
  NEXT

  GOTO Main

  SUB Wait1s
    CLRWDT
    PAUSE Approx1s
    CLRWDT
  END SUB
```



Data lines from USB are not connected.
We only want power.

NOTE:
D1 & D2 may be omitted
if the relays contain
snubber diodes

From:
http://fruitoftheshed.com/wiki/ - **FotS**

Permanent link:
http://fruitoftheshed.com/wiki/doku.php?id=great_cow_basic:hard_air_gap_a_device_to_isolate_portions_of_a_small_network

Last update: **2024/01/19 09:45**