PUZZLE 2 - PN532 ELECHOUE NFC MODULE

Ara programareu una versió gràfica del puzzle anterior, amb les biblioteques PyGObject i Ruby-GNOME2/gtk3. Hauríeu de fer servir el puzzle1 com a biblioteca. La finestra principal estará formada per un Label que demana login amb la targeta RF—carnet UPC o Mifare— i un Button de Clear. Quan es llegeix la targeta, apareix l'uid en hexadecimal:

M'he instal·lat les biblioteques de ruby gnomes i actualitzat el apt.

Tmb m'han fet falta les llibreries libnfc-dev i libfreefare-dev utilitzant sudo apt install libnfc-dev libfreefare-dev

Un cop s'executava bé la gran part del codi i ja es veia bé el label amb el uid, em vaig trobar que quan premia el botó de clear es borrava temporalment, ja que encara que apartessis la targeta tornava a aparèixer el uid impr`sa al label.
Això ho he solucionat netejant el uid, i assegurant-me que només hi hagi un fil a la vegada.

El codi del segón puzzle és la següent. Es pot comprovar que utilitza el puzzle anterior, el qual he necessitat fer-li algún retoc per poder-lo usar en aquest codi. Això ha fet que el batégés com ha vr2.rb, de versió 2.

```
require 'gtk3'
require 'thread'
require_relative 'vr2.rb'
BLUE = Gdk::RGBA.new(0, 0, 1, 0.65)
RED = Gdk::RGBA.new(1, 0, 0, 0.65)
WHITE = Gdk::RGBA.new(1, 1, 1, 1)
TEXTO_INICIAL = "Introdueix la targeta o clauer"

class Rfid_Reader
  def initialize(reader, handler)
    @reader = reader
    @handler = handler
  end

  def read_uid
    Thread.new do
      uid = @reader.read_uid

      uid = nil if uid.nil?
      GLib::Idle.add { @handler.call("#{uid}") }
    end
  end
end
window = Gtk::Window.new("Rfid_gtk.rb")
window.set_border_width(10)
```

```
window.set_position(Gtk::WindowPosition::CENTER)
window.signal_connect("delete_event") do
  Gtk.main_quit
  false
end
```

Aquest és un exemple d'execució.