#!/usr/bin/env python

# -\*- coding: utf8 -\*-

import RPi.GPIO as GPIO

import MFRC522

import signal

continue\_reading = True

# Capture SIGINT for cleanup when the script is aborted

def end\_read(signal,frame):

global continue\_reading

print "Ctrl+C captured, ending read."

continue\_reading = False

GPIO.cleanup()

# Hook the SIGINT

signal.signal(signal.SIGINT, end\_read)

# Create an object of the class MFRC522

MIFAREReader = MFRC522.MFRC522()

# Welcome message

print "Press Ctrl-C to stop."

# This loop keeps checking for chips. If one is near it will get the UID and authenticate

while continue\_reading:

# Scan for cards

(status,TagType) = MIFAREReader.MFRC522\_Request(MIFAREReader.PICC\_REQIDL)

# If a card is found

if status == MIFAREReader.MI\_OK:

print "Card detected"

# Get the UID of the card

(status,uid) = MIFAREReader.MFRC522\_Anticoll()

# If we have the UID, continue

if status == MIFAREReader.MI\_OK:

# Print UID

print "Card read UID: "+str(uid[0])+","+str(uid[1])+","+str(uid[2])+","+str(uid[3])

# This is the default key for authentication

key = [0xFF,0xFF,0xFF,0xFF,0xFF,0xFF]

# Select the scanned tag

MIFAREReader.MFRC522\_SelectTag(uid)

# Authenticate

status = MIFAREReader.MFRC522\_Auth(MIFAREReader.PICC\_AUTHENT1A, 8, key, uid)

# Check if authenticated

if status == MIFAREReader.MI\_OK:

MIFAREReader.MFRC522\_Read(8)

MIFAREReader.MFRC522\_StopCrypto1()

else:

print "Authentication error"