HackPSU RFID Scanner Box User Manual

September 28, 2018

Introduction

This document serves as the reference manual for the HackPSU RFID scanner box. This manual does not detail the implementation of the device; the intention of this document is to provide a user with the necessary information to operate the scanner and to troubleshoot some basic problems.

Contents

1 RFID Devices

This scanner uses the MFRC522 RFID transceiver unit. The present implementation is intended for use with MIFARE compatible RFID devices. This decision was made to simplify the implementation of the scanning functionality. Other devices may be compatible, but may not work.

2 Keypad Controls

The keypad behaves like a normal numeric keypad in most cases; however, the nonnumerical characters have special functions listed below.

A The 'A' key acts as a scroll up function in all menus.

B The 'B' key acts as a scroll down function in all menus, and the 'B' key acts as the back key when in a non-menu state, returning the user to the main menu.

C The 'C' key serves no special functionality presently. In a future implementation, it will scroll the LCD.

D The 'D' key transitions the box into the locked state.

***** The '*' key acts as a clear or negative input button.

The '#' key acts as a submit or positive input button.

3 Access Control

Access control is handled in a locked state, where the scanner waits for a staff wristband to be scanned. This is intended to provide a minimal layer of security on the device so that event goers cannot scan themselves into events. The duplicate state will produce a staff band by writing the master key into the data sector of the band. Access control can be disabled in the implementation.

4 Scanning

Upon entering the "Locate & Scan" state, the scanner will fetch a list of available locations from the cache server defined in the configuration header. To select an entry from the menu, press the '#' key. Numerical inputs are not supported in this menu. Once the location is selected; wristbands may be scanned. The scanner is set with a timeout, which cycles the functionality to listen for keypad driven state transitions using the 'B' and 'D' keys. Upon receiving a successful scan, a request is sent to the cache server to commit the scan. If the response is received and the "isRepeat" field is set to 'false' then the LCD will display "Allow" if no response is received or if the field is set to 'true' then "Deny" will be printed to the LCD. The only case where the "isRepeat" field will be set to 'true' is if the event at the current location is a 'food' event and the user has been scanned once before.

5 Check-in

The check-in state follows a linear flow. First a pin must be entered using the keypad. Once a pin is submitted, a request to retrieve the user data will be made. If no response is received or if the pin is not found on the cache server, the message ”Invalid Pin” will be displayed. Upon successfully entering a pin, a new prompt will appear, asking the user to validate the name displayed below. The user may press ’#’ to confirm that the user pin was correct, or ’*’ to return to pin entry. The user will then be prompted to scan a wristband. If the wristband’s UID already exists in the cache server’s database, then a message instructing the user to dispose of the band will be displayed. If the scan function times out before a wristband is scanned, the pin will need to be entered again. If the user name is confirmed and a wristband is scanned, the user will be checked-in on the cache server. The next step displays the user’s shirt size and asks for photo consent. Any key can be pressed to escape the photo consent menu, although using ’*’ or ’#’ is recommended as this may change in a future implementation. The scanner will then return to the beginning of the check-in process. Press ’B’ or ’D’ in the pin entry state to leave the check-in state.

6 Configuration

The configuration for the device must be done prior to deployment. All configuration should be done in the sketch’s hackPSUconfig.h file. The following fields are available:

MASTER_KEY This defines the key written to sector 4 of the staff wristbands used to unlock the scanner as well as the key that the duplicate state writes to staff bands.

SCAN_TIMEOUT The time that the scanner will block for when attempting to interact with a wristband.

BAUD_RATE The baud rate for the serial communications.

WIFI_CONSTS This definition should be empty; it is a flag marking whether or not the SSID and PASSWD fields have been defined; if unset, the sketch will fail to compile.

REDIS The address of the cache server; this should not have any protocol, just a network address and a port.

SSID The name of the network to connect to; if this network cannot be found, the scanner will not boot.

PASSWD The password for the network that is being connected to; if this is incorrect, the scanner will not boot.

fp The fingerprint of the SSL certificate being used by the caching server.

7 Troubleshooting

7.1 Long Scan Times

If scan times are taking a long time, the likely culprit is a bad RFID signal or a long RTT on the connection to the cache server. A bad RFID signal may be caused by excess material between the wristband and the scanner of the box. A long RTT can be caused by long network hops to the cache server or by network congestion. To test RTT with your cache server, use the following ”time_total” feature in cURL.

7.2 Failure to Boot

A failure to boot generally means that the network configuration is incorrect. Double check the network SSID and password are correct. If they are correct, the LCD should indicate that the network connection succeeded. The next blocking call will be to get an API key from the cache server. This call will fail if the cache server is misconfigured, or if the scanner is misconfigured. On the scanner side, check that the address is correct and does not have ”http://” or ”https://” prepended to the hostname. Also validate that the port on the server is correct. The next point of failure will be in the SSL certificate fingerprint or if HTTPS is not implemented on the server, the implementation for the API wrapper must be changed to use HTTP instead of HTTPS.

If the box fails reaches the point where it is locked and will not enter the menu state, the user must scan a wristband, which contains the master key value as defined in the configuration header. If your wristbands do not support read/write access, undefine SECURE_BOOT in the state machine header to bypass a secure boot.

7.3 Unexpected Shutdowns

The scanner may crash and shutdown. When this happens, the most common failure mode during developent was an unhanded exception like a memory allocation failure. The scanner will attempt to reenter the program. In this case, unplug the power and force the scanner to boot from the beginning.

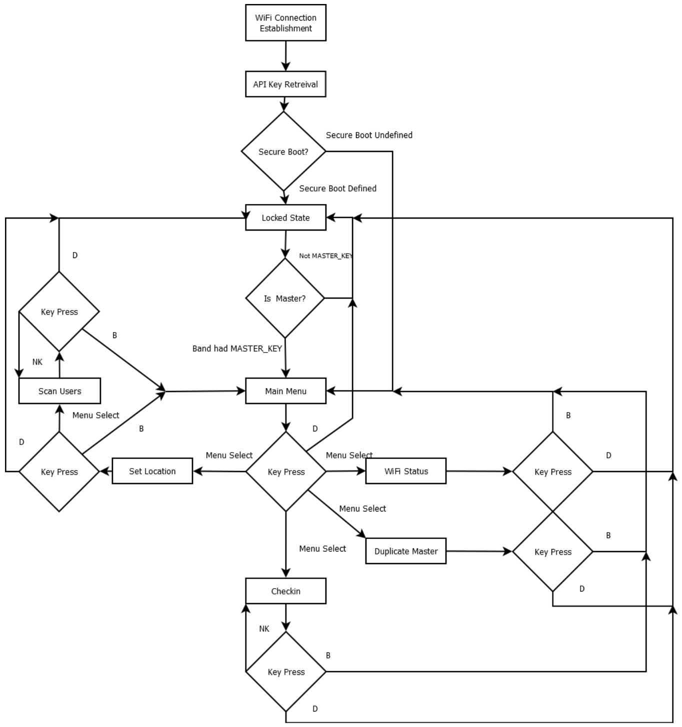
If the scanner shuts down unexpectedly, check its battery charge; the device does not know how much battery is left and will not give a low battery warning.

7.4 Soft Locked Scanner

The scanner should not ever soft lock. In the event that it does, the best option is to power cycle the device. Another alternative would be to hold the ’B’ or ’D’ key and hope that the box times out and transitions to the menu or lock state.

8 Appendix

8.1 Finite State Machine


WiFi Connection Establishment
API Key Retreival
Secure Boot?
Secure Boot Undefined
Secure Boot Defined
Locked State
Not MASTER_KEY
Is Master?
Band had MASTER_KEY
Main Menu
D
Key Press
B
NK
Scan Users
Menu Select
B
D
Key Press
Set Location
Menu Select
Key Press
D
Menu Select
WiFi Status
Menu Select
Duplicate Master
Menu Select
Checkin
B
Key Press
D
Key Press
B
D
NK
Key Press
B
D