Ø=Ý

SenseStamp

Proof-of-Concept Hardware Specification

ESP32-C6 + PN532 NFC + Battery

DaSecure Solutions LLC · San Francisco, CA
Document Version 1.0 · January 31, 2026
CONFIDENTIAL

1. Project Overview

SenseStamp is an IoT security platform delivering tiny wireless sensors with instant push notifications — no hub, no fees, no complexity. This document specifies the Proof-of-Concept (POC) hardware build combining an ESP32-C6 microcontroller with a PN532 NFC/RFID module and LiPo battery power.

Primary POC Goal: Proof of Presence — cryptographically prove a specific NFC tag was scanned at a specific location at a specific time, with instant cloud notification.

Target Applications

Ø=ÜÍ Proof of Presence

Scan NFC tag at location to cryptographically verify presence at a specific time. Security patrols, delivery confirmation, maintenance rounds.

Ø=Ý Smart Seal / Tamper Tag

NFC tag on doors, safes, packages. Periodic presence check — tag removed triggers instant push notification.

Ø=Þª Tap-to-Arm/Disarm

NFC card tap toggles security zone. No app needed — hotel key card experience for your security.

Ø<ß÷þ Product Authentication

Write signed hash to NFC tags on high-value items. Any NFC phone can verify authenticity.

Ø=Üd Check-in System

Each person gets an NFC card. Tap on arrival/departure. Logs who, where, when automatically.

Ø=Ý' Tool Checkout

Tagged tools + sensor at tool crib. Tap in/out tracks inventory, alerts if items don't return.

2. System Architecture

```
% % % % % % % % % % % % % % %      SPI Bus     % % % % % % % % % % % % % % % % % % %      Wi-Fi      % % % %
%              %   SCK/MISO/MOSI   %                  %     802.11n    %               %
%    PN532     % %Ä% % % % % % % % % % % % % % % % %º%     ESP32-C6     % %Ä% % % % % % % % % % % % %º%   Clou
%   NFC/RFID   %      + SS/IRQ     %   Microcontroller%               %   (iotpush)   %
%              %                   %                  %               %               %
% % % % % % % % % % % % % % %                    % % % % % % % % % %,% % % % % % % % %                % % % % %
                                          %                                %
                                    % % % % % % %4% % % % % %                  % % % % % % %4% % % % % % % %
                                    %   TP4056   %                    %     Push     %
                                    %   Charger  %                    %   Notification%
                                    %   + LiPo   %                    %    to Phone  %
                                    % % % % % % % % % % % % %                  % % % % % % % % % % % % % % %
```

```
FLOW:  NFC Tag Tap !’ PN532 IRQ !’ ESP32 Wake !’ Read Tag UID !’ Sign Timestamp
       !’ Connect Wi-Fi !’ POST to Cloud API !’ Push Notification !’ Deep Sleep
```

3. Hardware Components

3.1 Bill of Materials

Component	Model / Spec	Qty	Est. Cost
Microcontroller	ESP32-C6 Dev Board (Seeed XIAO)	1	$6–8
NFC Module	PN532 NFC/RFID Module	1	$4–6
Battery	3.7V LiPo, 1000mAh	1	$3–5
Charger Module	TP4056 USB-C w/ protection	1	$0.50–1
Capacitor	100;Äb VÆV7G&öÇ—F–0	1	$0.10
Breadboard	Half-size 400 tie-point	1	$2
Jumper Wires	Male-to-female dupont	~10	$1
NFC Tags	NTAG215 stickers	5–10	$3–5
TOTAL			$20–28

3.2 ESP32-C6 Specifications

Parameter	Value
CPU	RISC-V 32-bit, single-core, 160 MHz
Wi-Fi	802.11b/g/n (2.4 GHz), Wi-Fi 6 ready
Bluetooth	BLE 5.0 + Zigbee / Thread (802.15.4)
Flash / SRAM	4 MB / 512 KB
GPIO	22 programmable GPIOs
Deep Sleep Current	~5 ;Ä
Active Current (Wi-Fi TX)	~300 mA peak
Operating Voltage	3.0–3.6V

3.3 PN532 NFC Specifications

Parameter	Value
NFC Standards	ISO 14443A/B, ISO 18092, FeliCa
Supported Tags	MIFARE Classic/Ultralight, NTAG2xx
Interfaces	SPI, I2C, UART (DIP switch selectable)
Operating Voltage	3.3V – 5V
Active / Standby Current	~100 mA / ~80 ;Ä
Read Range	~5 cm (tag dependent)
Frequency	13.56 MHz

4. Wiring Diagram

4.1 PN532 !’ ESP32-C6 (SPI Mode)

```
 PN532 NFC Module                          ESP32-C6 Dev Board
% % % % % % % % % % % % % % % % % % %                       % % % % % % % % % % % % % % % % % % % % % % %
%                  %                       %                      %
%   SCK  % % % % % % % % % % %<% % % % % % % % % % % % % % % % % % % % % % %$%º GPIO 6   (SPI CLK) %
%                  %                       %                      %
%   MISO % % % % % % % % % % %<% % % % % % % % % % % % % % % % % % % % % % %$%º GPIO 2   (SPI RX)  %
%                  %                       %                      %
%   MOSI % % % % % % % % % % %<% % % % % % % % % % % % % % % % % % % % % % %$%º GPIO 7   (SPI TX)  %
%                  %                       %                      %
%   SS   % % % % % % % % % % %<% % % % % % % % % % % % % % % % % % % % % % %$%º GPIO 10  (SPI CS)  %
%                  %                       %                      %
%   IRQ  % % % % % % % % % % %<% % % % % % % %  (optional) % % %$%º GPIO 3   (WAKE)    %
%                  %                       %                      %
%   VCC  % % % % % % % % % % %<% % % % % % % % % % % % % % % % % % % % % % %$%º 3V3                %
%                  %                       %                      %
%   GND  % % % % % % % % % % %<% % % % % % % % % % % % % % % % % % % % % % %$%º GND                %
% % % % % % % % % % % % % % % % % % %                       % % % % % % % % % % % % % % % % % % % % % % %

DIP Switches (PN532):  SW1: OFF  |  SW2: ON  !’  SPI Mode
```

4.2 Power Circuit

```
                             % % % % % % % % % % % % % % % % % % %
% % % % % % % % % % % %                  %     TP4056       %
%   3.7V    %                  %   USB-C Charger  %
%   LiPo    %                  %   w/ Protection  %
%  1000mAh  %                  %                  %            % % % % % % % % % % % % % % % %
%   BAT+ % % % %<% % % % % % % % % % % % % % % % % %$%º B+          OUT+ % % % % % % % % % %$%º 3V3 / VIN
%           %                  %                  %     % % % %    %    ESP32-C6   %
%   BAT- % % % %<% % % % % % % % % % % % % % % % % %$%º B-          OUT- % % % %$  % % % % %$%º GND
%           %                  %                  %     % % % %    %               %
% % % % % % % % % % % %                  %    USB-C IN      %    100;Äb  QBP P P P P P P P P P P P P P Q
                             % % % % % % % % % % % % % % % % % % %
```

& þ Voltage Warning: Both PN532 and ESP32-C6 operate at 3.3V logic. No level shifting needed. Do NOT connect 5V to PN532 VCC.

4.3 Full Pin Map

ESP32-C6 Pin	Connected To	Function
GPIO 6	PN532 SCK	SPI Clock
GPIO 2	PN532 MISO	SPI Master In, Slave Out
GPIO 7	PN532 MOSI	SPI Master Out, Slave In
GPIO 10	PN532 SS	SPI Chip Select (active low)
GPIO 3	PN532 IRQ	Interrupt for deep sleep wake

3V3	PN532 VCC, TP4056 OUT+	3.3V power rail
GND	PN532 GND, TP4056 OUT-	Common ground

5. Firmware Architecture

5.1 Boot & Runtime Flow

```
% % % % % % % % % % %      % % % % % % % % % % % %      % % % % % % % % % % % % %      % % % % % % % % % % %
%   DEEP   %      %   WAKE    %      %   READ     %      %   SIGN &   %      %   SEND    %
%   SLEEP  % % % % % %º%   on IRQ  % % % % % %º%   NFC TAG  % % % % % %º%   TIMESTAMP% % % % % %º%   TO API  %
%   (~5;Ä ’ P"     P"  ÷" F–ÖW"P"     P"  T”B      P"     P"  „„Ô 2’   P"     P"  …v’Ôf’’ P
% % % % % % % % % % %      % % % % % % % % % % % %      % % % % % % % % % % % % %      % % % % % % % % % % %
     %²                                                                    %
     % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % % %
                                  confirm & return to deep sleep
```

5.2 Key Features

- Deep Sleep with IRQ Wake: ESP32-C6 sleeps at ~5;Ä à PN532 IRQ triggers wake on tag detect.
- Tag UID Reading: Read NFC tag unique identifier (4 or 7 byte UID) via SPI.
- Cryptographic Signing: HMAC-SHA256 of {tag_uid + device_id + timestamp}.
- Wi-Fi Connect & POST: Connect to Wi-Fi, send signed payload to cloud API.
- LED Feedback: Brief flash — green for success, red for failure.
- Timer Wake (Optional): Periodic wake for tamper-check mode.

5.3 Proof-of-Presence Payload

```
{
  "device_id":  "ss-001-c6",
  "tag_uid":    "04:A2:3B:7C:D1:00:00",
  "timestamp":  1738346100,
  "location":   "warehouse-door-3",
  "signature":  "a3f8c2...e91b",
  "battery_mv": 3720,
  "fw_version": "0.1.0"
}
```

5.4 Development Stack

Option	Pros	Recommendation
Arduino + ESP32 Core	Fast prototyping, PN532 libs available	POC '
ESP-IDF (C)	Full control, lower power	Production
PlatformIO	Better tooling, same libraries	POC '

5.5 Required Libraries

- Adafruit_PN532 — NFC/RFID communication
- WiFi (built-in) — ESP32-C6 Wi-Fi stack
- HTTPClient (built-in) — HTTPS POST to cloud
- mbedtls (built-in) — HMAC-SHA256 signing

- ArduinoJson — JSON payload construction

6. Power Analysis

6.1 Current Consumption

State	ESP32-C6	PN532	Total	Duration
Deep Sleep	5 ;Ä	80 ;Ä	~85 ;Ä	99.5% of time
Wake + NFC Read	30 mA	100 mA	~130 mA	~200 ms
Wi-Fi TX	300 mA	80 ;Ä	~300 mA	~2 sec
Wi-Fi Connect	120 mA	80 ;Ä	~120 mA	~1 sec

6.2 Battery Life Estimate

```
Scenario: 20 NFC scans/day, 1000mAh battery
Sleep: 85;Ä  r #2ã“†, Ò "ã B Ô ,öF y
Active: 20 scans × 3.2s × 200mA = 3.6 mAh/day
Total: ~5.6 mAh/day !’ ~178 days (~6 months)
With MOSFET PN532 power cut: 12+ months
```

7. Security Considerations

- Device Key: Unique pre-shared key in ESP32-C6 flash (eFuse for production).
- HMAC Signing: Every event signed with HMAC-SHA256, preventing replay/forgery.
- TLS: All API communication over HTTPS with certificate pinning.
- Tag Cloning: POC uses UID-based. Production: NTAG 424 DNA with AES authentication.
- Timestamp Integrity: NTP sync on wake; RTC drift monitoring; server ±30s validation.

8. Cloud Integration

The POC integrates with iotpush (DaSecure's push notification platform) for the notification pipeline:

- ESP32-C6 POSTs signed JSON payload to iotpush API endpoint
- iotpush validates signature and routes notification
- Push notification delivered to mobile app and/or webhook
- Event stored in database for audit trail / proof retrieval

9. POC Milestones

Phase	Milestone	Description
1	Hardware Assembly	Wire ESP32-C6 + PN532 on breadboard, verify SPI
2	NFC Read	Read NFC tag UIDs, display on serial monitor
3	Wi-Fi + API	Connect to Wi-Fi, POST tag event to cloud
4	Crypto Signing	HMAC-SHA256 proof-of-presence payload
5	Deep Sleep	IRQ-based wake, power optimization
6	Battery Integration	LiPo + TP4056, voltage monitoring
7	End-to-End Demo	Tap tag !’ push notification in <3 seconds

10. Next Steps (Post-POC)

- Custom PCB design (KiCad) — reduce form factor to stamp-size
- 3D printed enclosure with adhesive mount
- NTAG 424 DNA support for clone-proof authentication
- BLE provisioning (instead of hardcoded Wi-Fi credentials)
- OTA firmware updates via cloud
- SenseStamp mobile app for tag management and event history
- Thread/Zigbee mesh support (ESP32-C6 native) for multi-sensor deployments