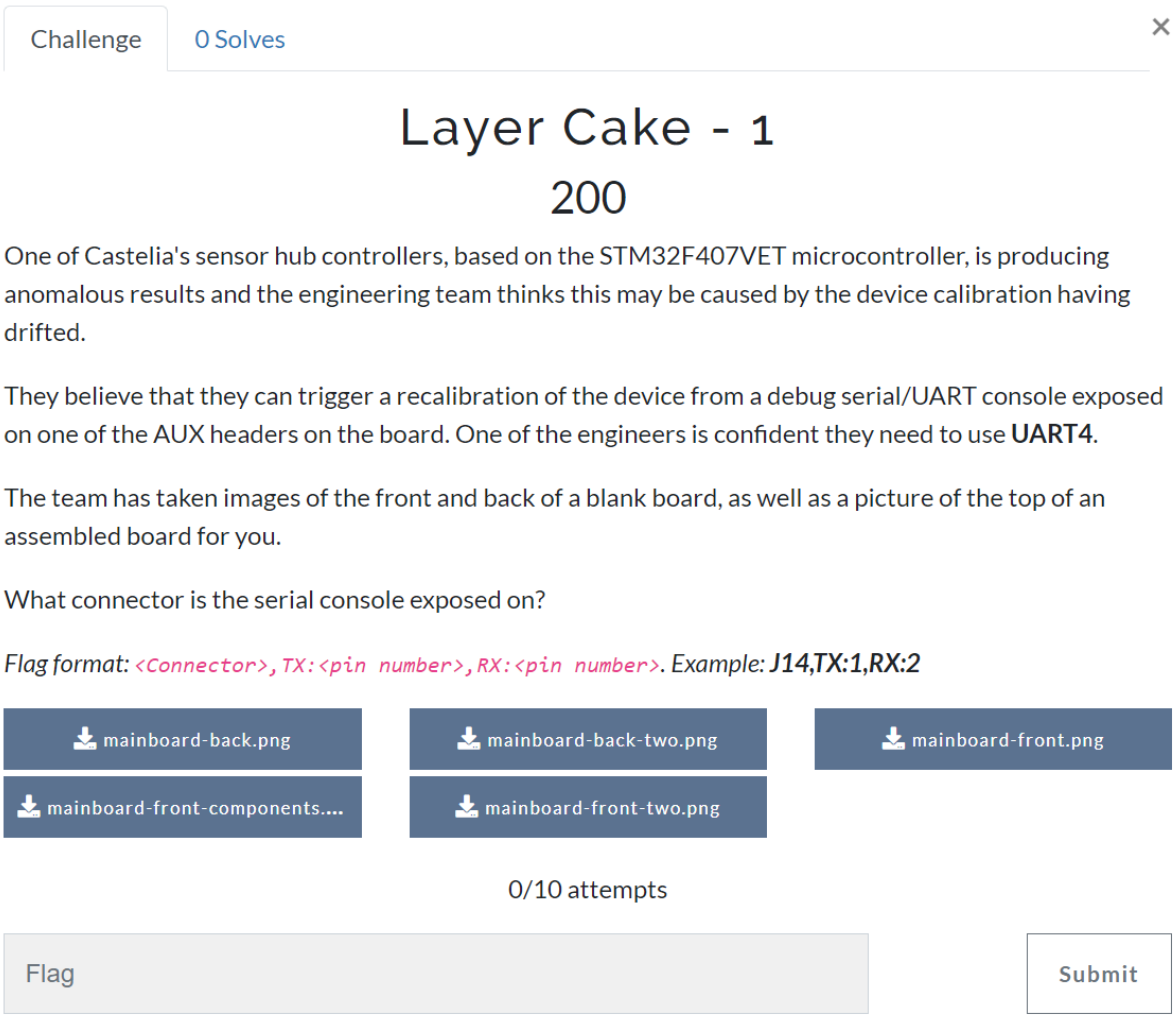
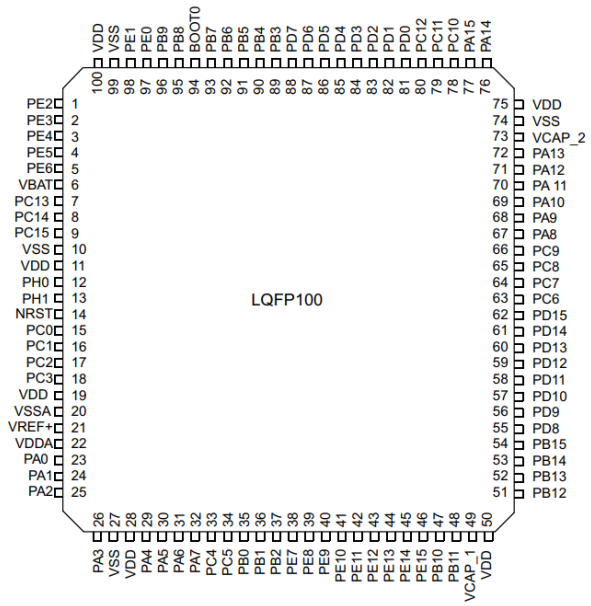
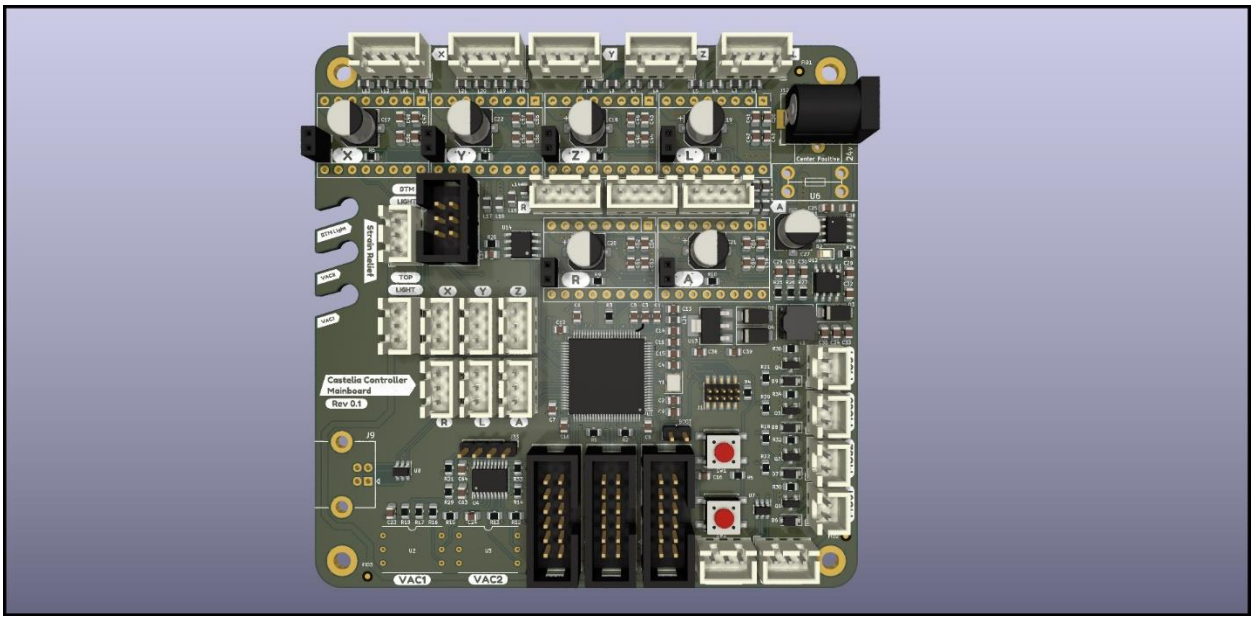
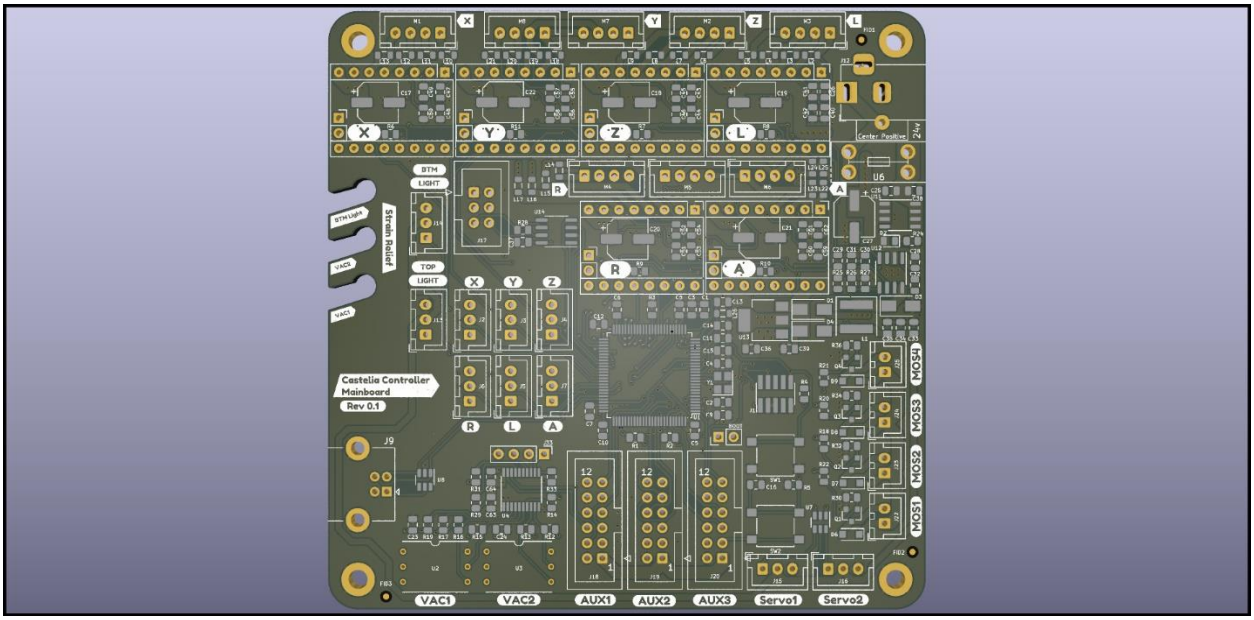
Layer Cake 1 challenge

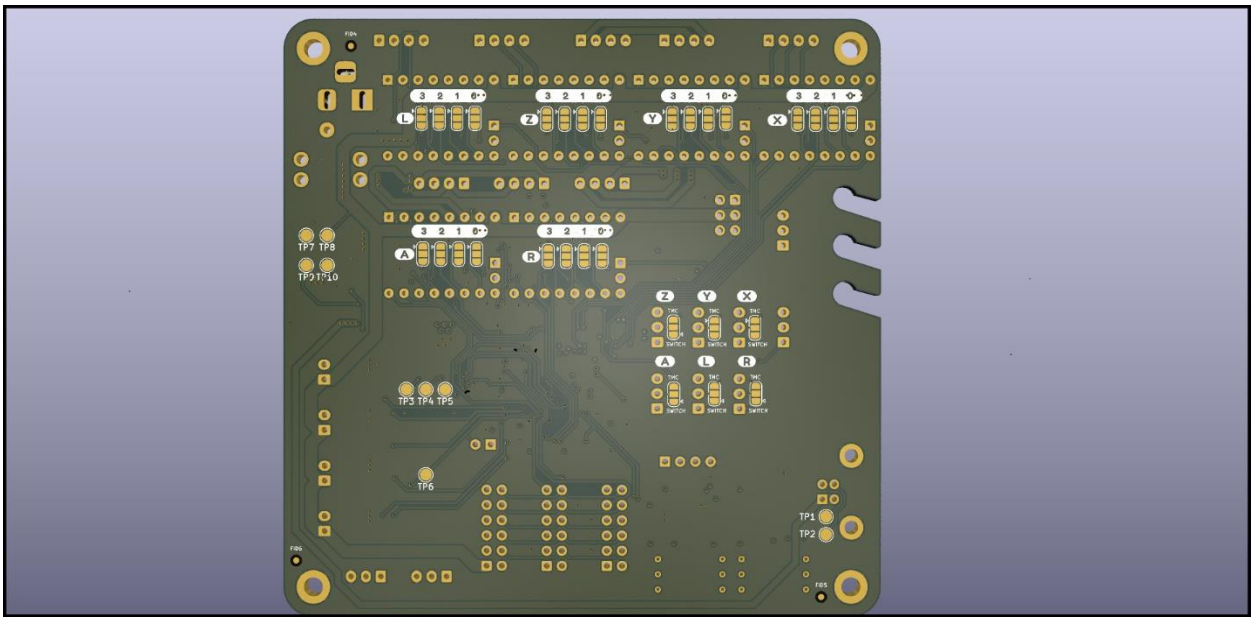


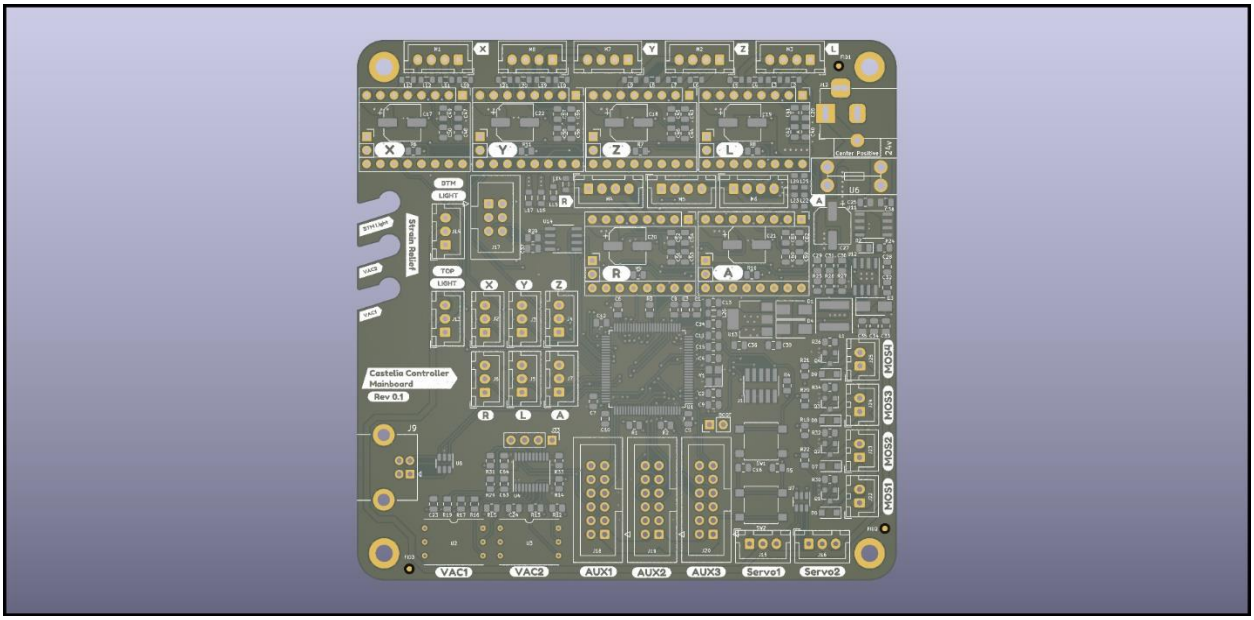
In the Layer Cake 1 challenge, I located the datasheet for the main controller and found that it uses an LQFP100 package. While the microcontroller features multiple UART channels, our focus is on UART4. According to the datasheet, the TX and RX pins for UART4 are assigned to pins 23 and 24, respectively. I traced these pins on the multi-layer PCB and found that the traces lead to connector J20. Specifically, the RX pin is connected to pin 11, and the TX pin is connected to pin 9.

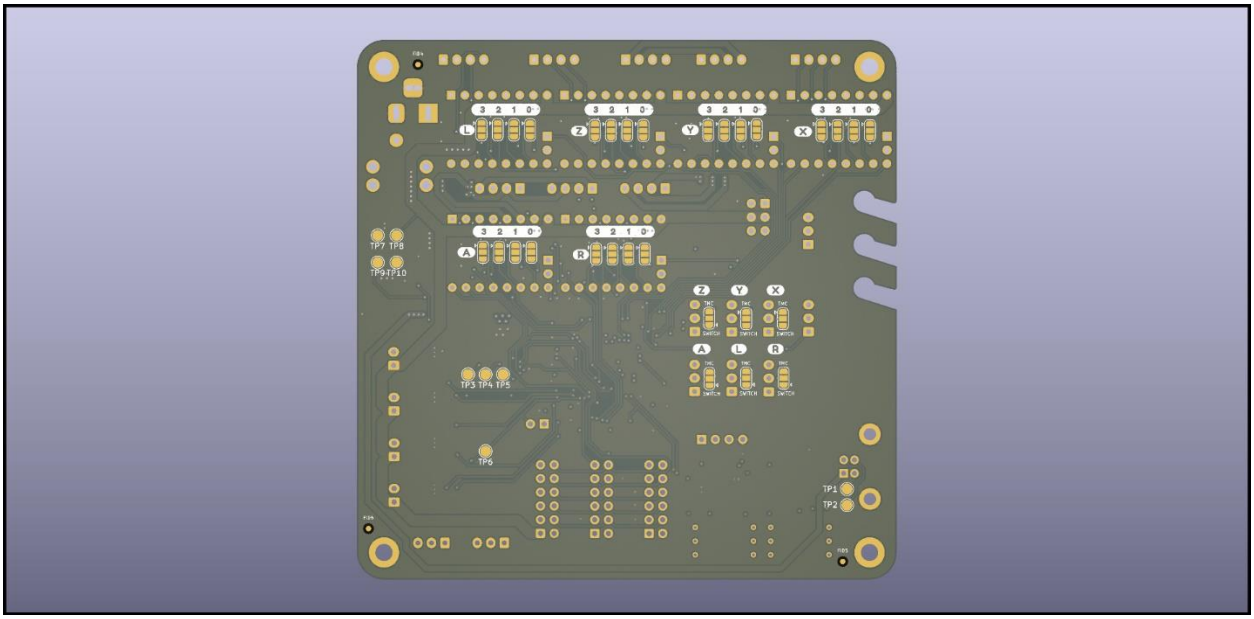

LQFP100
1 PE2
2 PE3
3 PE4
4 PE5
5 PE6
6 VBAT
7 PC13
8 PC14
9 PC15
10 VSS
11 VDD
12 PH0
13 PH1
14 NRST
15 PC0
16 PC1
17 PC2
18 PC3
19 VDD
20 VSSA
21 VREF+
22 VDDA
23 PA0
24 PA1
25 PA2
26 PA3
27 VSS
28 VDD
29 PA4
30 PA5
31 PA6
32 PA7
33 PC4
34 PC5
35 PB0
36 PB1
37 PB2
38 PE7
39 PE8
40 PE9
41 PE10
42 PE11
43 PE12
44 PE13
45 PE14
46 PE15
47 PB10
48 PB11
49 VCAP_1
50 VDD
51 PB12
52 PB13
53 PB14
54 PB15
55 PD8
56 PD9
57 PD10
58 PD11
59 PD12
60 PD13
61 PD14
62 PD15
63 PC6
64 PC7
65 PC8
66 PC9
67 PA8
68 PA9
69 PA10
70 PA 11
71 PA12
72 PA13
73 VCAP_2
74 VSS
75 VDD
76 PA14
77 PA15
78 PC10
79 PC11
80 PC12
81 PD0
82 PD1
83 PD2
84 PD3
85 PD4
86 PD5
87 PD6
88 PD7
89 PB3
90 PB4
91 PB5
92 PB6
93 PB7
94 BOOT0
95 PB8
96 PB9
97 PE0
98 PE1
99 VSS
100 VDD


Castelia Controller
Mainboard
Rev 0.1
VAC1
VAC2

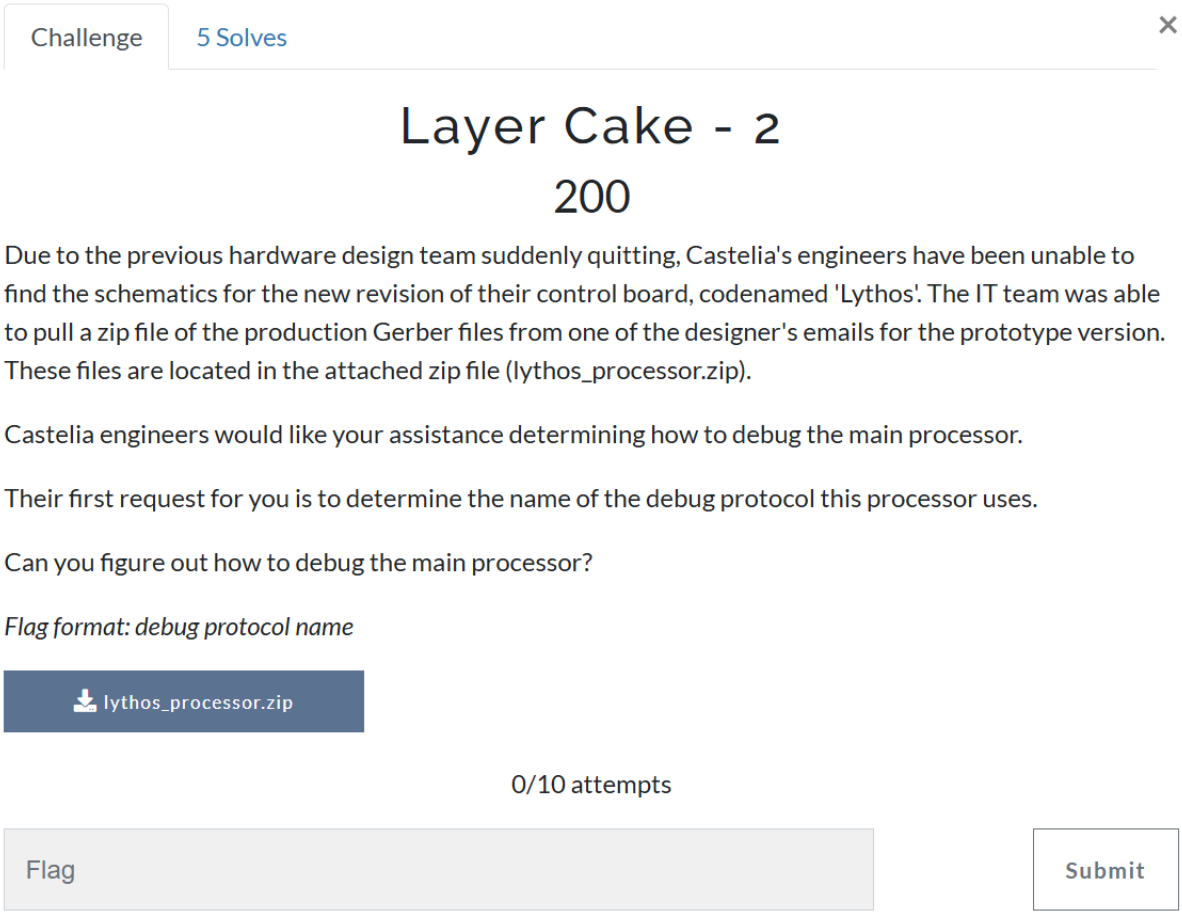

Castelia Controller
Mainboard
Rev 0.1
Strain Relief
BTM LIGHT
TOP LIGHT
Center Positive
VAC1
VAC2
AUX1
AUX2
AUX3
Servo1
Servo2


TP1
TP2
TP3 TP4 TP5
TP6
TP7 TP8
TP9 TP10

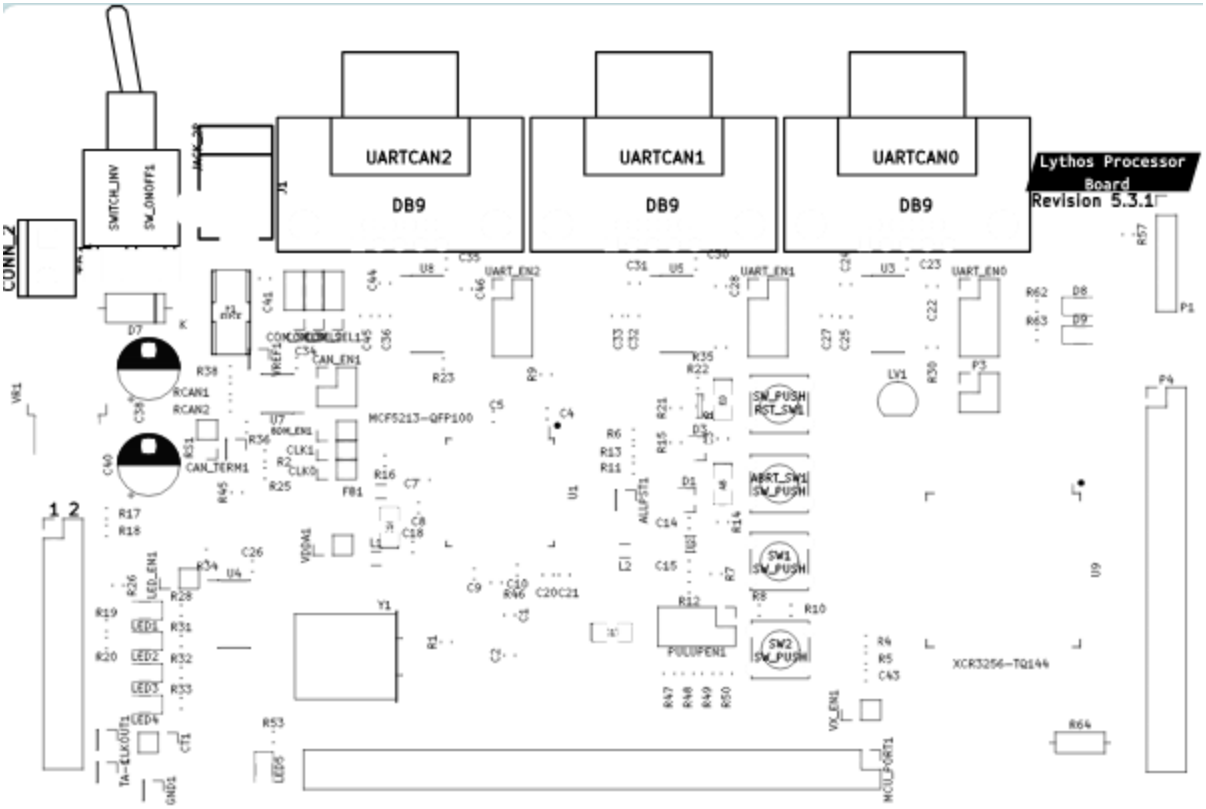

Castelia Controller
Mainboard
Rev 0.1
VAC1
VAC2
AUX1
AUX2
AUX3
Servo1
Servo2

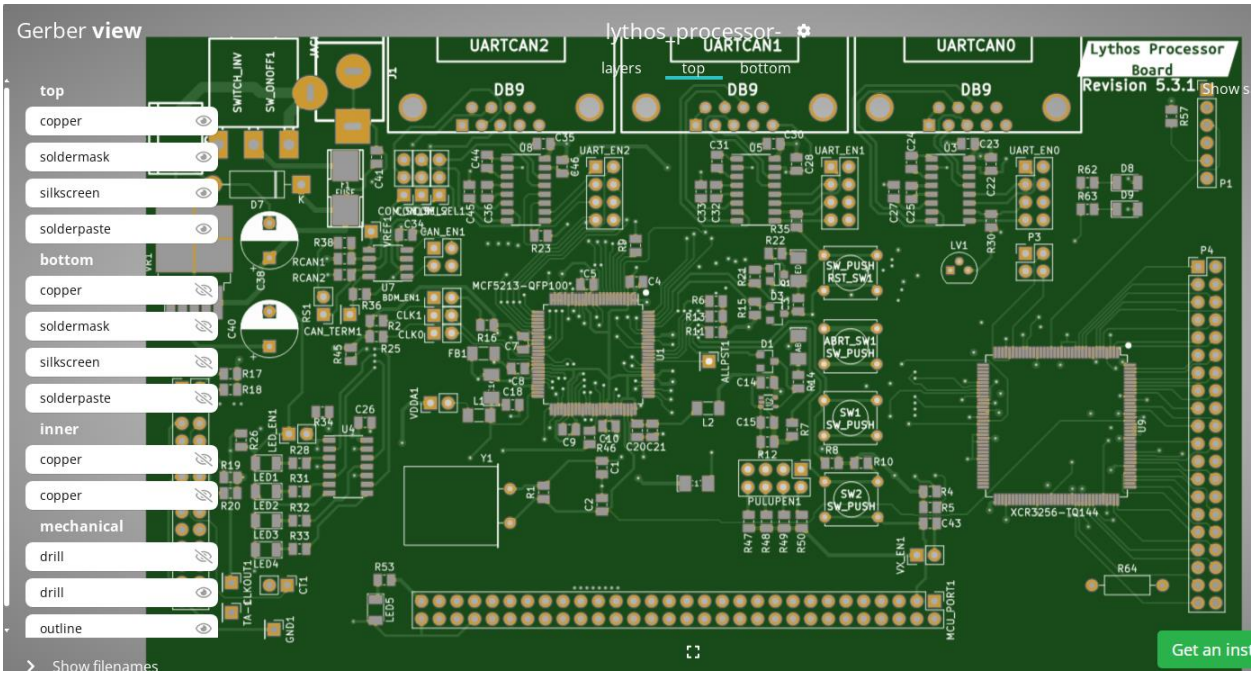

TP7 TP8
TP9 TP10
TP3 TP4 TP5
TP6
TP1
TP2

Layer Cake 2 challenge



The lythos_processor-F_Silkscreen.gto file appears to be a Gerber file, commonly used in PCB design to define the graphical elements of printed circuit boards. This file specifically contains details about the PCB's layers, apertures, and drawing instructions for the silkscreen (the top legend layer in this case). I used the Online Gerber Viewer tool from PCBWay to identify the processor name. The microcontroller is labeled MCF5213 - QFP100. After searching for the name online, I found its datasheet, which revealed the debug protocol used: BDM (Background Debug Mode).

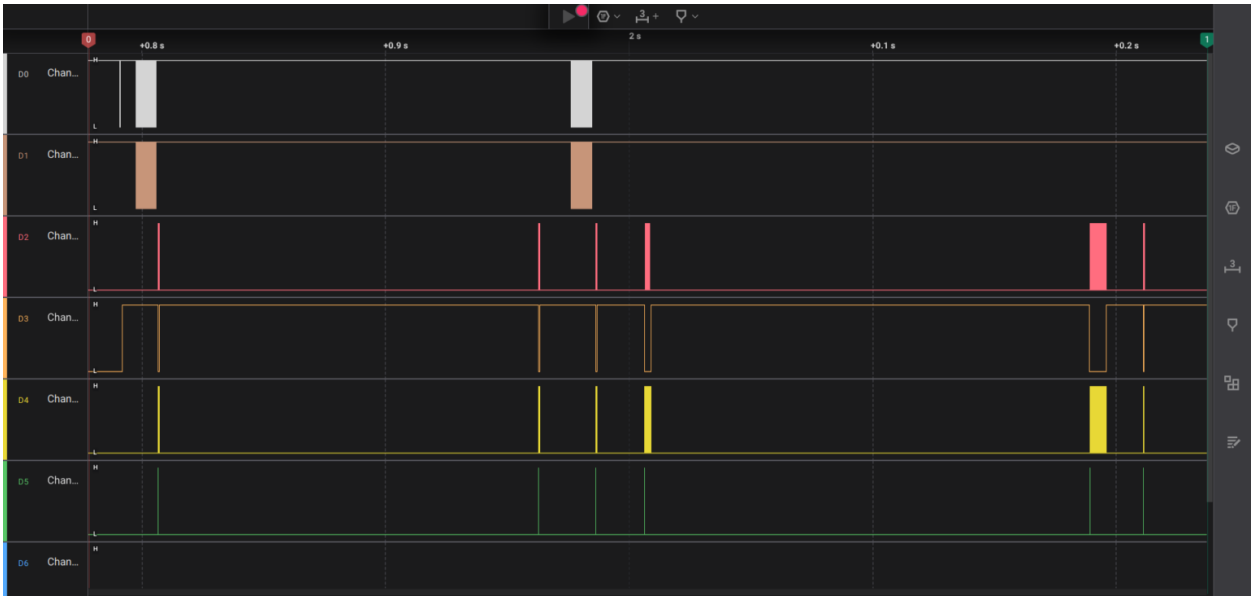

UARTCAN2
DB9
UARTCAN1
DB9
UARTCAN0
DB9
Lythos Processor Board
Revision 5.3.1
SWITCH_INV
SW_ONOFF1
MCF5213-QFP100
XCR3256-TQ144
SW_PUSH
RST_SW1
ABRT_SW1
SW1
SW2
CAN_TERM1
PULUPEN1
UART_EN2
UART_EN1
UART_EN0
CAN_EN1
LED1
LED2
LED3
LED4
LED5
TA-CLKOUT1
MCU_PORT1

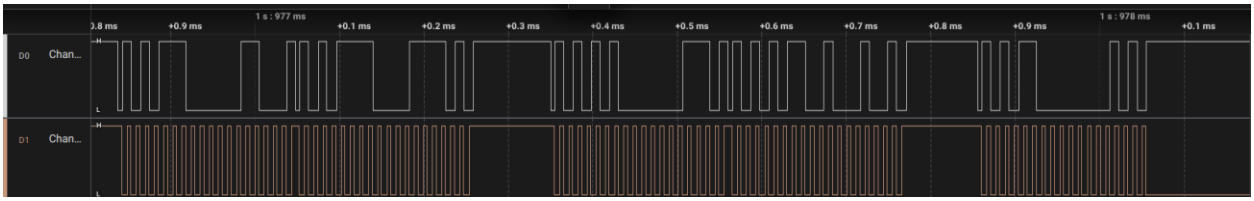

Gerber view
lythos_processor-
layers
top
bottom
top
copper
soldermask
silkscreen
solderpaste
bottom
copper
soldermask
silkscreen
solderpaste
inner
copper
copper
mechanical
drill
drill
outline
Show filenames
UARTCAN2
DB9
UARTCAN1
DB9
UARTCAN0
DB9
Lythos Processor Board
Revision 5.3.1
Show s
Get an inst

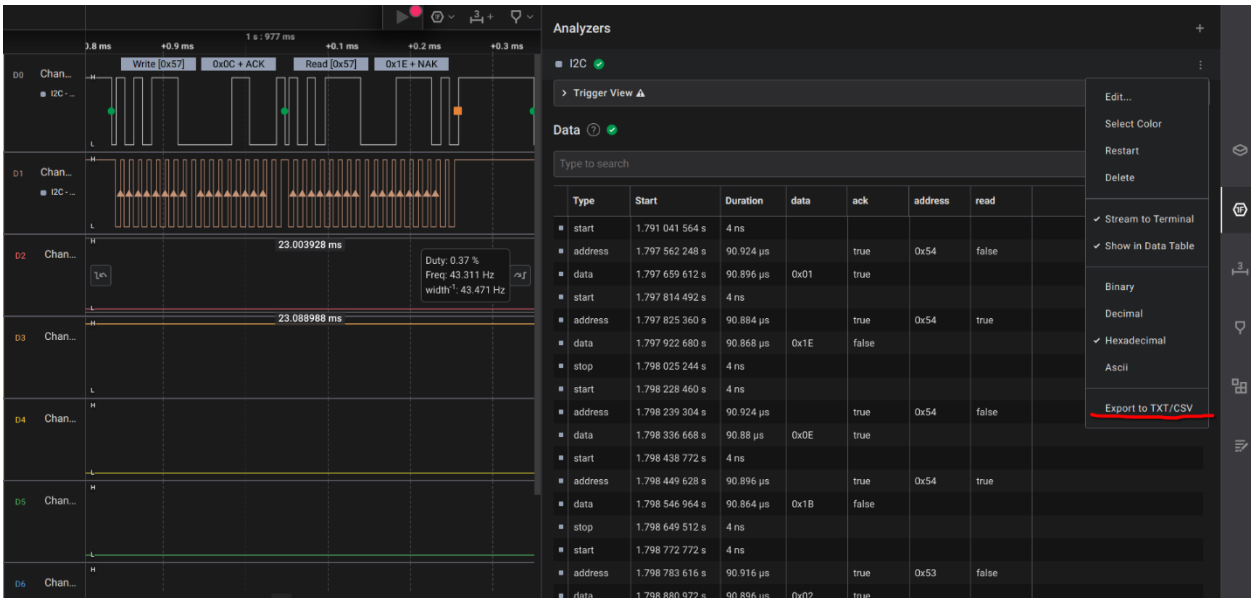
Spy By Wire 1

We analyzed the memory.sal file using Saleae's Logic Analyzer software. The 24LC001 EEPROM communicates via the I2C protocol, a widely used communication standard, particularly for microcontrollers interfacing with peripheral ICs on the same PCB. I2C operates with two lines: the SDA (data) line and the SCL (clock) line. The SCL provides a periodic clock signal, while the SDA transmits data, which should only change when SCL is low. Given this, we deduced that SCL is likely mapped to Channel 1 and SDA to Channel 0. Using the Logic Analyzer, we applied an I2C protocol filter to extract all relevant data in .csv format. We then sorted the data into read and write operations. During the sorting process, we observed that the data repeats in 16-byte sections.

Flag: 0x691E5E4219581B44190C1F421E471B58

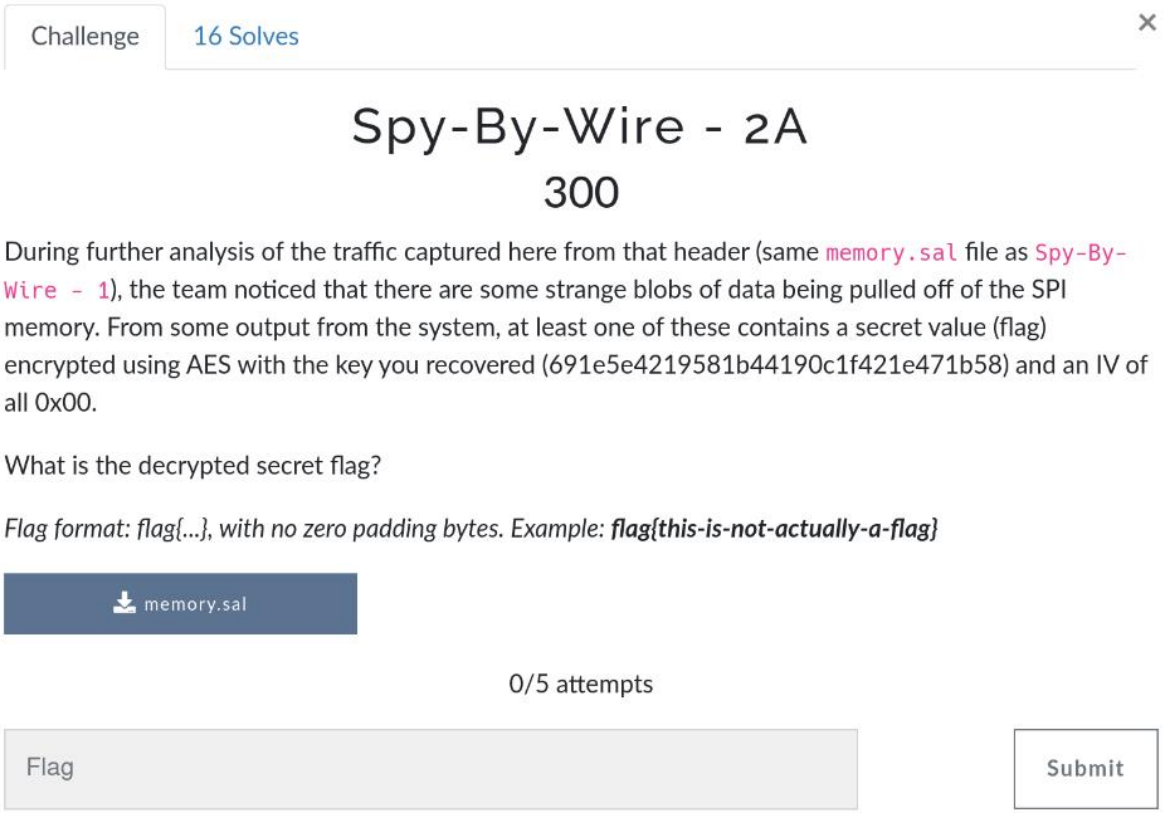

+0.8 s
+0.9 s
2 s
+0.1 s
+0.2 s
D0 Chan...
D1 Chan...
D2 Chan...
D3 Chan...
D4 Chan...
D5 Chan...
D6 Chan...


1 s : 977 ms
1 s : 978 ms
D0 Chan...
D1 Chan...


Write [0x57]
0x0C + ACK
Read [0x57]
0x1E + NAK
23.003928 ms
23.088988 ms
Duty: 0.37 %
Freq: 43.311 Hz
width⁻¹: 43.471 Hz
Analyzers
I2C
Trigger View
Data
Type to search
Edit...
Select Color
Restart
Delete
Stream to Terminal
Show in Data Table
Binary
Decimal
Hexadecimal
Ascii
Export to TXT/CSV
Type | Start | Duration | data | ack | address | read
start | 1.791 041 564 s | 4 ns
address | 1.797 562 248 s | 90.924 µs | | true | 0x54 | false
data | 1.797 659 612 s | 90.896 µs | 0x01 | true
start | 1.797 814 492 s | 4 ns
address | 1.797 825 360 s | 90.884 µs | | true | 0x54 | true
data | 1.797 922 680 s | 90.868 µs | 0x1E | false
stop | 1.798 025 244 s | 4 ns
start | 1.798 228 460 s | 4 ns
address | 1.798 239 304 s | 90.924 µs | | true | 0x54 | false
data | 1.798 336 668 s | 90.88 µs | 0x0E | true
start | 1.798 438 772 s | 4 ns
address | 1.798 449 628 s | 90.896 µs | | true | 0x54 | true
data | 1.798 546 964 s | 90.864 µs | 0x1B | false
stop | 1.798 649 512 s | 4 ns
start | 1.798 772 772 s | 4 ns
address | 1.798 783 616 s | 90.916 µs | | true | 0x53 | false

write	to	0x54	ack	data:	0x01		0x69	0x00
read	to	0x54	ack	data:	0x1E		0x1E	0x01
write	to	0x54	ack	data:	0x0E		0x5E	0x02
read	to	0x54	ack	data:	0x1B		0x42	0x03
write	to	0x53	ack	data:	0x02		0x19	0x04
read	to	0x53	ack	data:	0x5E		0x58	0x05
write	to	0x52	ack	data:	0x05		0x1B	0x06
read	to	0x52	ack	data:	0x58		0x44	0x07
write	to	0x57	ack	data:	0x03		0x19	0x08
read	to	0x57	ack	data:	0x42		0x0C	0x09
write	to	0x56	ack	data:	0x07		0x1F	0x0A
read	to	0x56	ack	data:	0x44		0x42	0x0B
write	to	0x51	ack	data:	0x08		0x1E	0x0C
read	to	0x51	ack	data:	0x19		0x47	0x0D
write	to	0x51	ack	data:	0x0A		0x1B	0x0E
read	to	0x51	ack	data:	0x1F		0x58	0x0F
write	to	0x54	ack	data:	0x00			
read	to	0x54	ack	data:	0x69			
write	to	0x50	ack	data:	0x0B			
read	to	0x50	ack	data:	0x42			
write	to	0x54	ack	data:	0x0C			
read	to	0x54	ack	data:	0x1E			
write	to	0x51	ack	data:	0x0F			
read	to	0x51	ack	data:	0x58			
write	to	0x53	ack	data:	0x0D			
read	to	0x53	ack	data:	0x47			
write	to	0x50	ack	data:	0x04			
read	to	0x50	ack	data:	0x19			
write	to	0x56	ack	data:	0x06			
read	to	0x56	ack	data:	0x1B			
write	to	0x56	ack	data:	0x09			

Spy By Wire 2



SPI typically uses four signals: a Clock signal, two data lines (MISO and MOSI), and an Enable signal. While this is the most common SPI configuration, other variants exist. We filtered the MOSI, MISO, Enable, and Clock channels based on the signal sources. After reviewing the filtered data, we exported it to a .csv file.Upon inspection, we observed that MOSI and MISO alternate communication, though MOSI transmits more data than MISO. There were six exchanges between them, but only one contains the output we need. To proceed, we first need to convert the hex data from the first exchange—after cleaning it of "0x" prefixes and spaces—into bytes using CyberChef. Finally, we will decrypt the bytes using AES.

Key: 691e5e4219581b44190c1f421e471b58

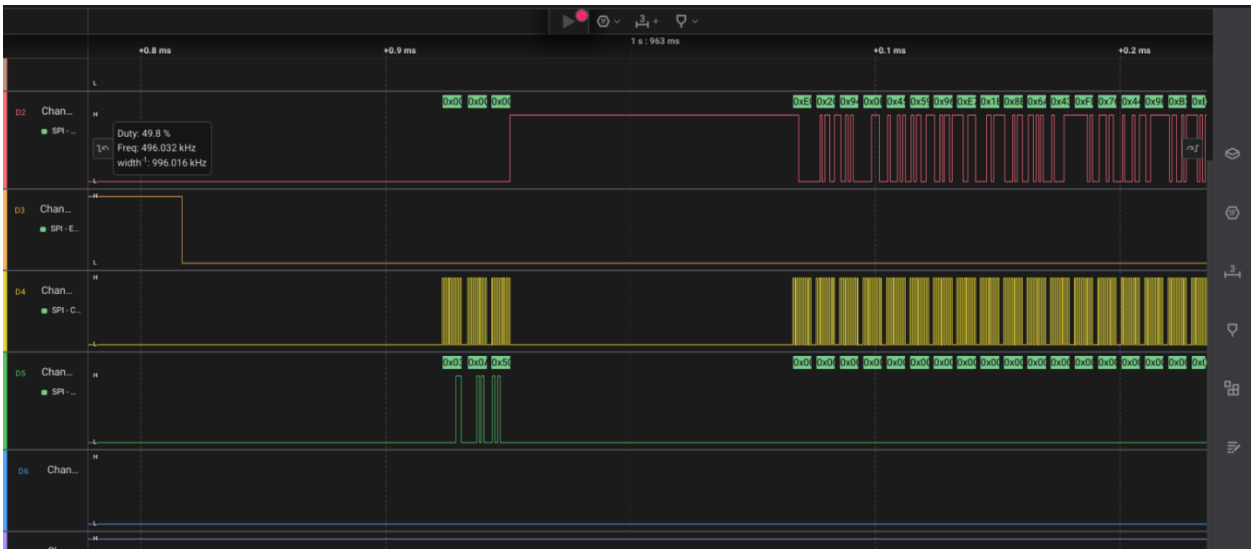
Mode: CBC

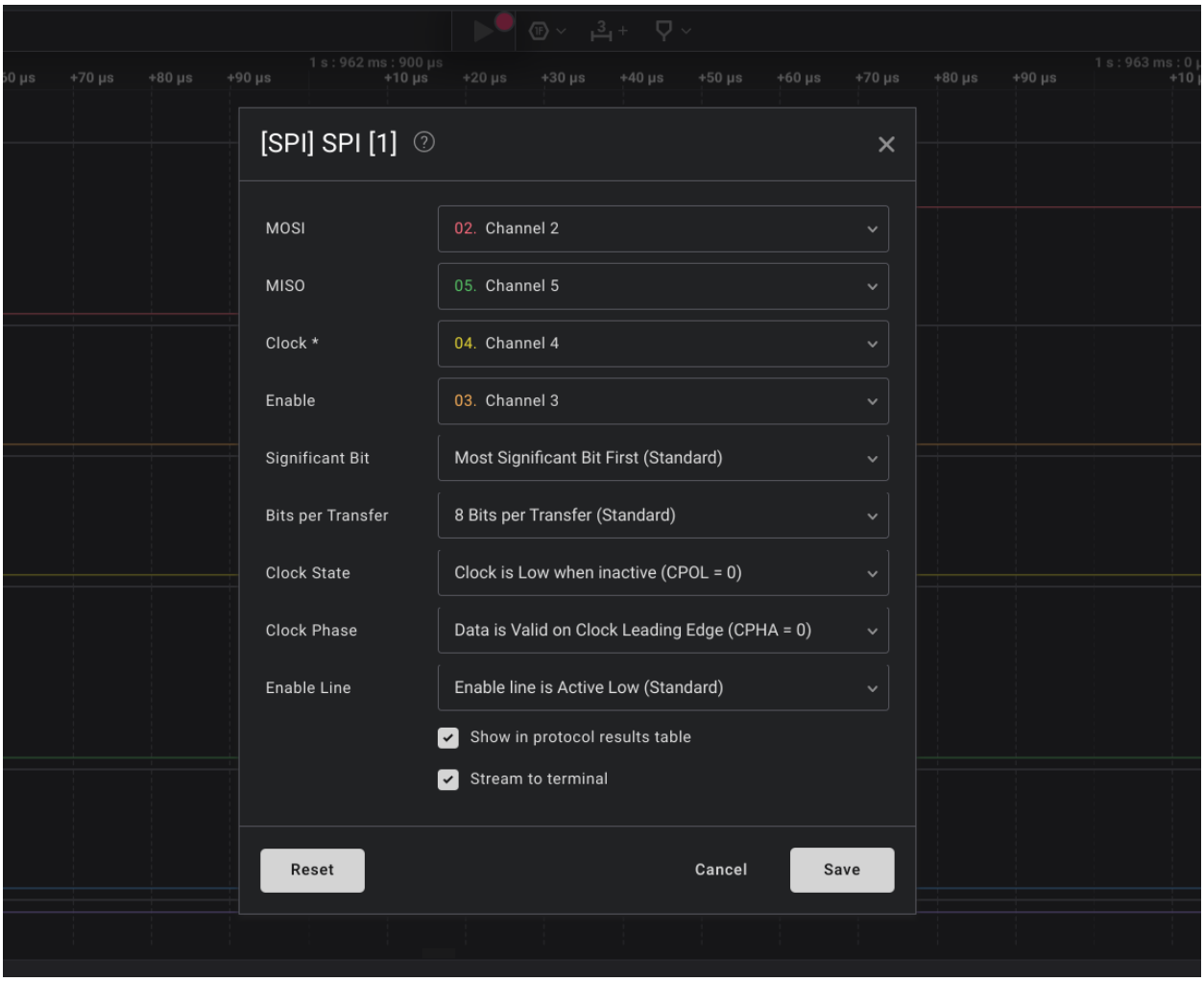
IV: 00000000000000000000000000000000

Clean Hex Data:

D9F8520E73DE46E72783BD2A7F8EF86CC34FEB63AE0A100A264CDBB33B41B59C

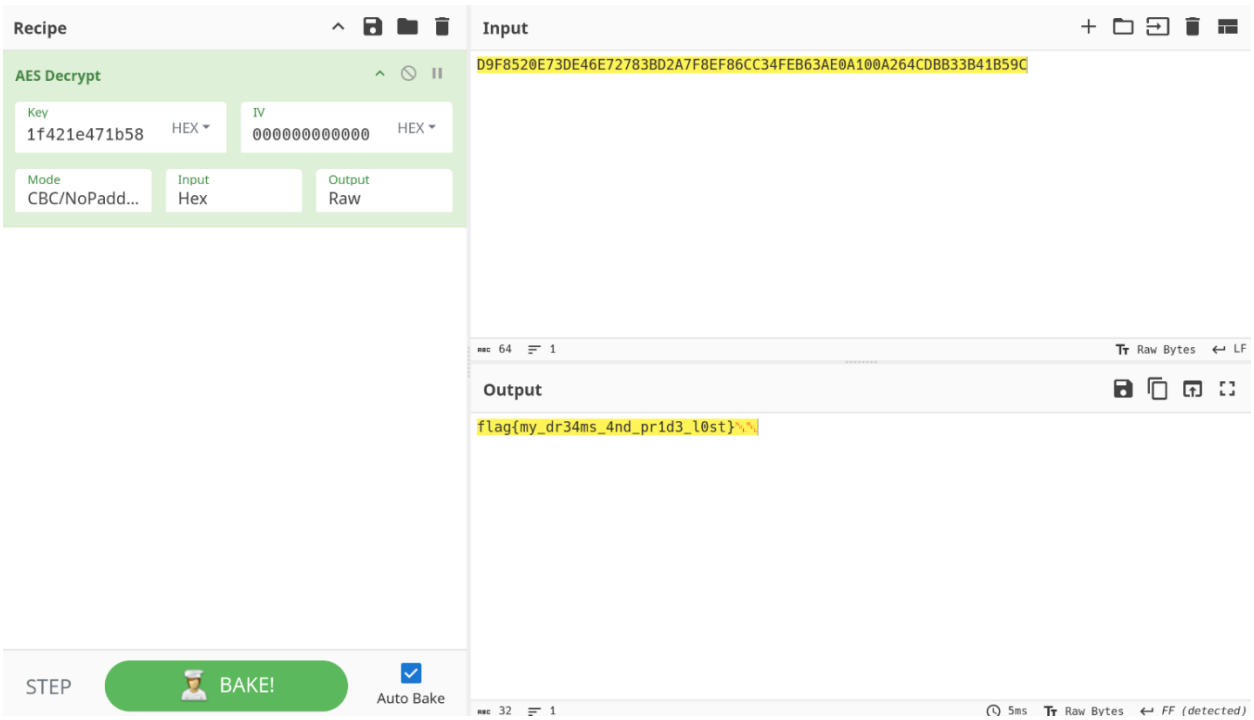
Flag: flag{my_dr34ms_4nd_pr1d3_l0st}





Time [s]	Time [s]	MOSI	MISO
1.806671504	0	0x00	0x03
1.806683432	0	0x00	0x0A
1.806693008	0	0x00	0x2F
1.80684196	0	0xD9	0x00
1.806851536	0	0xF8	0x00
1.806861112	0	0x52	0x00
1.806870688	0	0x0E	0x00
1.806880264	0	0x73	0x00
1.80688984	0	0xDE	0x00
1.806899416	0	0x46	0x00
1.806908992	0	0xE7	0x00
1.806918568	0	0x27	0x00
1.806928144	0	0x83	0x00
1.80693772	0	0xBD	0x00
1.806947296	0	0x2A	0x00
1.806956872	0	0x7F	0x00
1.806966448	0	0x8E	0x00
1.806976024	0	0xF8	0x00
1.8069856	0	0x6C	0x00
1.806995176	0	0xC3	0x00
1.807004752	0	0x4F	0x00
1.807014328	0	0xEB	0x00
1.807023904	0	0x63	0x00
1.80703348	0	0xAE	0x00
1.807043056	0	0x0A	0x00
1.807052632	0	0x10	0x00
1.807062208	0	0x0A	0x00
1.807071784	0	0x26	0x00
1.80708136	0	0x4C	0x00
1.807090936	0	0xDB	0x00

First	Second	Third	Fourth	Fifth	Sixth
0xD9	0xE0	0xD9	0x62	0x8B	0xE0
0xF8	0x2C	0xF8	0x3D	0xE2	0x2C
0x52	0x94	0x52	0x62	0xD8	0x94
0x0E	0x0E	0x0E	0x79	0x1F	0x0E
0x73	0x45	0x73	0x74	0x52	0x45
0xDE	0x59	0xDE	0x65	0xEC	0x59
0x46	0x96	0x46	0x73	0x08	0x96
0xE7	0xE7	0xE7	0x0A	0xC6	0xE7
0x27	0x1B	0x27	0x73	0x2C	0x1B
0x83	0x8B	0x83	0x68	0xD3	0x8B
0xBD	0x6A	0xBD	0x61	0x42	0x6A
0x2A	0x43	0x2A	0x2E	0x13	0x43
0x7F	0xFD	0x7F	0x75	0x6B	0xFD
0x8E	0x76	0x8E	0x70	0x1C	0x76
0xF8	0x44	0xF8	0x64	0x7D	0x44
0x6C	0x9F	0x6C	0x61	0x59	0x9F
0xC3	0xB2	0xC3	0x74	0xED	0xB2
0x4F	0xEB	0x4F	0x65	0xDD	0xEB
0xEB	0x1C	0xEB	0x28	0x1E	0x1C
0x63	0x7B	0x63	0x6B	0xC0	0x7B
0xAE	0x1D	0xAE	0x2B	0xFD	0x1D
0x0A	0x17	0x0A	0x62	0xF9	0x17
0x10	0x74	0x10	0x27	0x07	0x74
0x0A	0xAE	0x0A	0x62	0x18	0xAE
0x26	0xDE	0x26	0x6C	0xFB	0xDE
0x4C	0x84	0x4C	0x75	0x69	0x84
0xDB	0x53	0xDB	0x65	0x85	0x53
0xB3	0x71	0xB3	0x73	0x78	0x71
0x3B	0x70	0x3B	0x6B	0x4A	0x70
0x41	0xA1	0x41	0x79	0x37	0xA1
0xB5	0x1E	0xB5	0x73	0xD9	0x1E
0x9C	0xDB	0x9C	0x61	0x5F	0xDB
			0x6E	0xFA	
			0x64	0xB3	
			0x61	0xE2	
			0x62	0xBD	


Recipe
AES Decrypt
Key
1f421e471b58
HEX
IV
000000000000
HEX
Mode
CBC/NoPadd...
Input
Hex
Output
Raw
STEP
BAKE!
Auto Bake
Input
D9F8520E73DE46E72783BD2A7F8EF86CC34FEB63AE0A100A264CDBB33B41B59C
64
1
Raw Bytes
LF
Output
flag{my_dr34ms_4nd_pr1d3_l0st}
32
1
5ms
Raw Bytes
FF (detected)