### **firelizzard *—* 08/14/2023 8:12 PM**

I think you can impersonate the badgiimos pretty easily with an Arduino. I'll put the details in a thread to avoid spoilers. **Spoilers for encryption, impersonation, and ? badgiimo challenges.**

### **firelizzard *—* 08/14/2023 8:14 PM**

When we attached a badgiimo with the analyzer connected, we saw the following sequence on the I2C bus:

* The badge writes "BDGIIMO?" to address 0x1F
* The badge requests 8 bytes from address 0x1F
* The badgiimo responds with it's ID encrypted with HHV{5up3 (the first 8 bytes of the encryption key)

The badgiimos and their IDs are:

* Game badgiimo (ID 1) sends IIWz4tq2
* Party badgiimo (ID 2) sends JJTy7wr1
* Call badgiimo (ID 3) sends KKUx6vs0
* Uber badgiimo (ID 4) sends LLR\x7F1qt7

IIWz4tq2 is 11111111 xored with HHV{5up3

### **firelizzard *—* 08/14/2023 8:23 PM**

We were able to impersonate the call badgiimo simply by responding to the badge's I2C request with KKUx6vs0, and the uber badgiimo by responding with LLR\x7F1qt7

? badgiimo (the uber badgiimo) required a bit more work to fully solve, as did encryption. I'm happy to explain how we solved those too (in a different thread) if anyone is interested.

(this message exists to hide the spoilers)

====================================================================

### **firelizzard *—* 08/16/2023 10:58 PM**

The first thing I'd do is convert the assembly to 'normal' pseudocode

I'm going to call the temp register T for brevity

MOVLW 0x5A ; W = 0x5A

MOVWF TEMP\_REG ; T = W

MOVLW 0xF8 ; W = 0xF8

ANDWF TEMP\_REG,1 ; T = W & T

XORWF TEMP\_REG,1 ; T = W ^ T

ADDWF TEMP\_REG,0 ; W = W + T

### **9lyph *—* 08/16/2023 11:01 PM**

Yep, started doing off down that path I couldn't seem to get the correct DC, C and Z register values

I have recently assembled to hex file and am looking for a simulator to handle those registers and spit out the required ouput

\*output

But I like how you simplified back to psuedo code, allot cleaner that my mock up 🙂

This is what I got

### **firelizzard *—* 08/16/2023 11:04 PM**

W = 0x5A # W = 0x5A = 0101 1010

T = W # T = 0x5A = 0101 1010

W = 0xF8 # W = 0xF8 = 1111 1000

T = W & T # T = 0x58 = 0101 1000

T = W ^ T # T = 0xA0 = 1010 0000

W = W + T # W = 0x98, carry 1

C is 1 because the lap step carried

Z is 0 because the result of the last step was not zero

I don't really know what the DC flag is. Something about the first nibble. So maybe it's supposed to be zero because that nibble didn't carry? Either way, there are only two possibilities so it's easy to test both and 0 works.

### **=============================================================**