DEFCON CHINA 1.0 BADGE HACKING WORKSHOP

JOE GRAND AKA KINGPIN

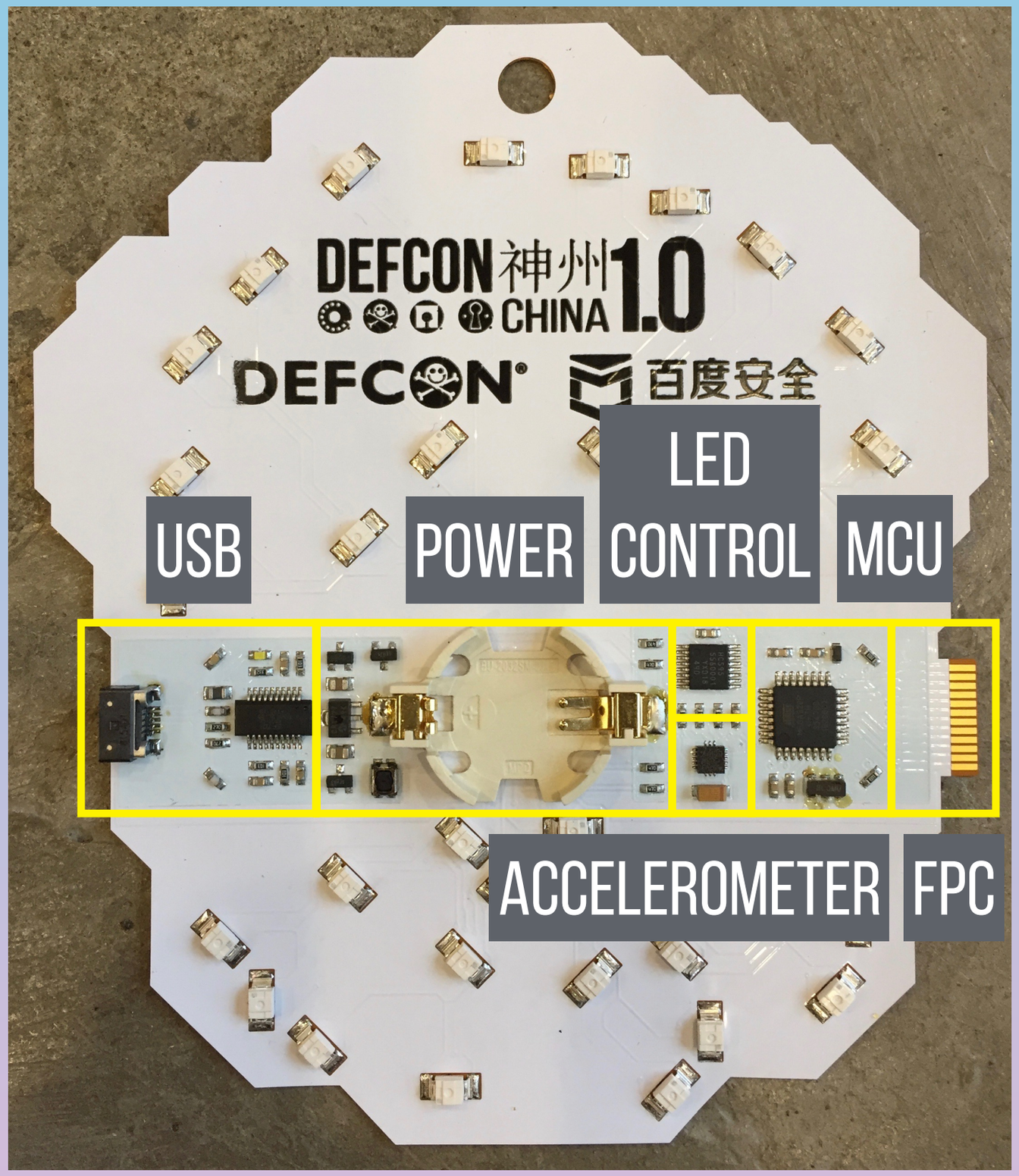
WORKSHOP GOALS

- INFORMAL ENVIRONMENT
- A DEEPER LOOK INTO THE BADGE
- SETUP DEVELOPMENT ENVIRONMENT
- MODIFY/RECOMPILE CODE
- OPEN LAB

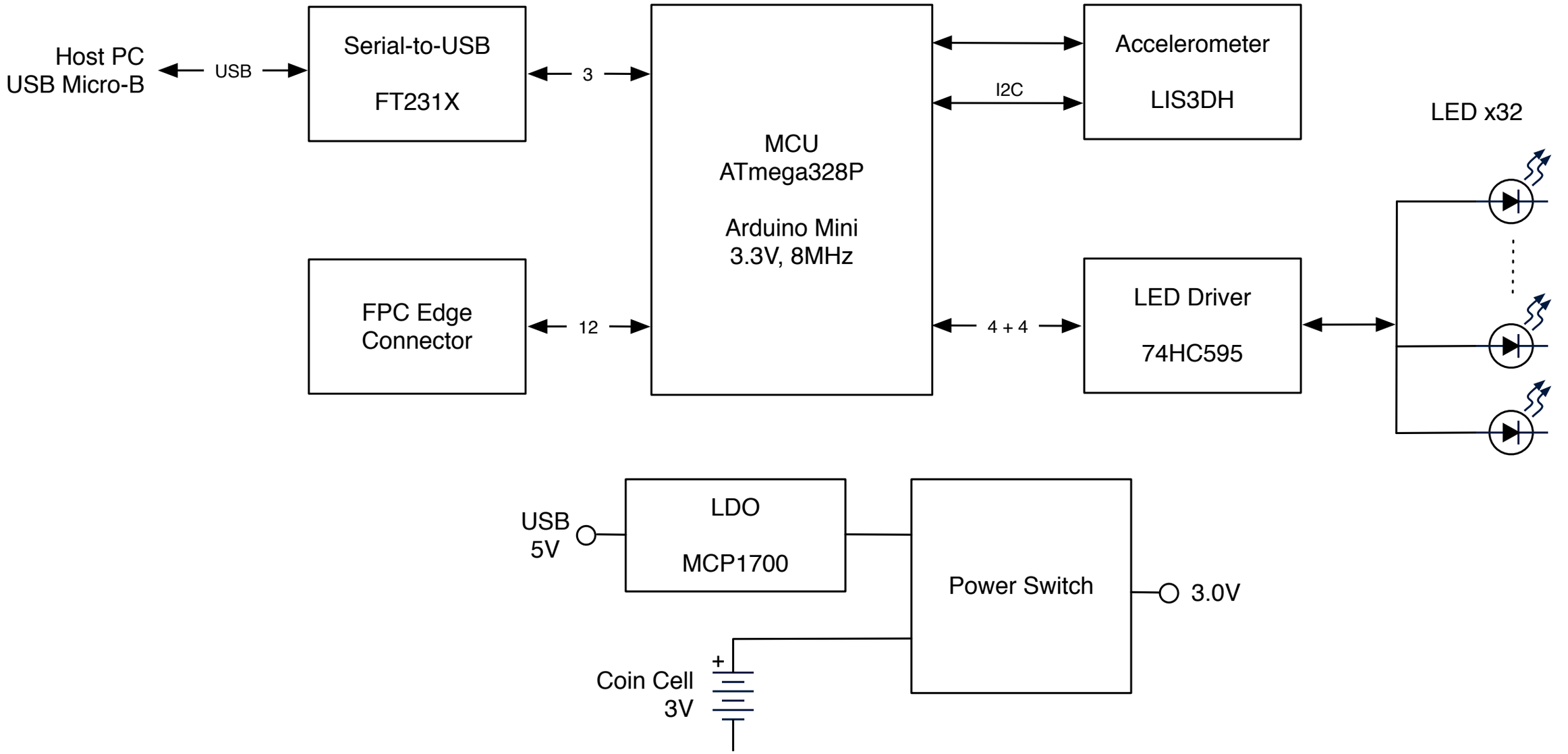
GAMEPLAY

- COMPLETE TASKS, GET REWARDED
- 4 ROOTS AND 4 BRANCHES, EACH WITH 4 LEDS
- WHEN TASK IS COMPLETE, BADGE INSERTED INTO PROGRAMMER TO UNLOCK LED
- WHEN EACH ROOT IS COMPLETE, MAGIC HAPPENS
- WHEN ALL ROOTS ARE COMPLETE, EVEN MAGIC HAPPENS

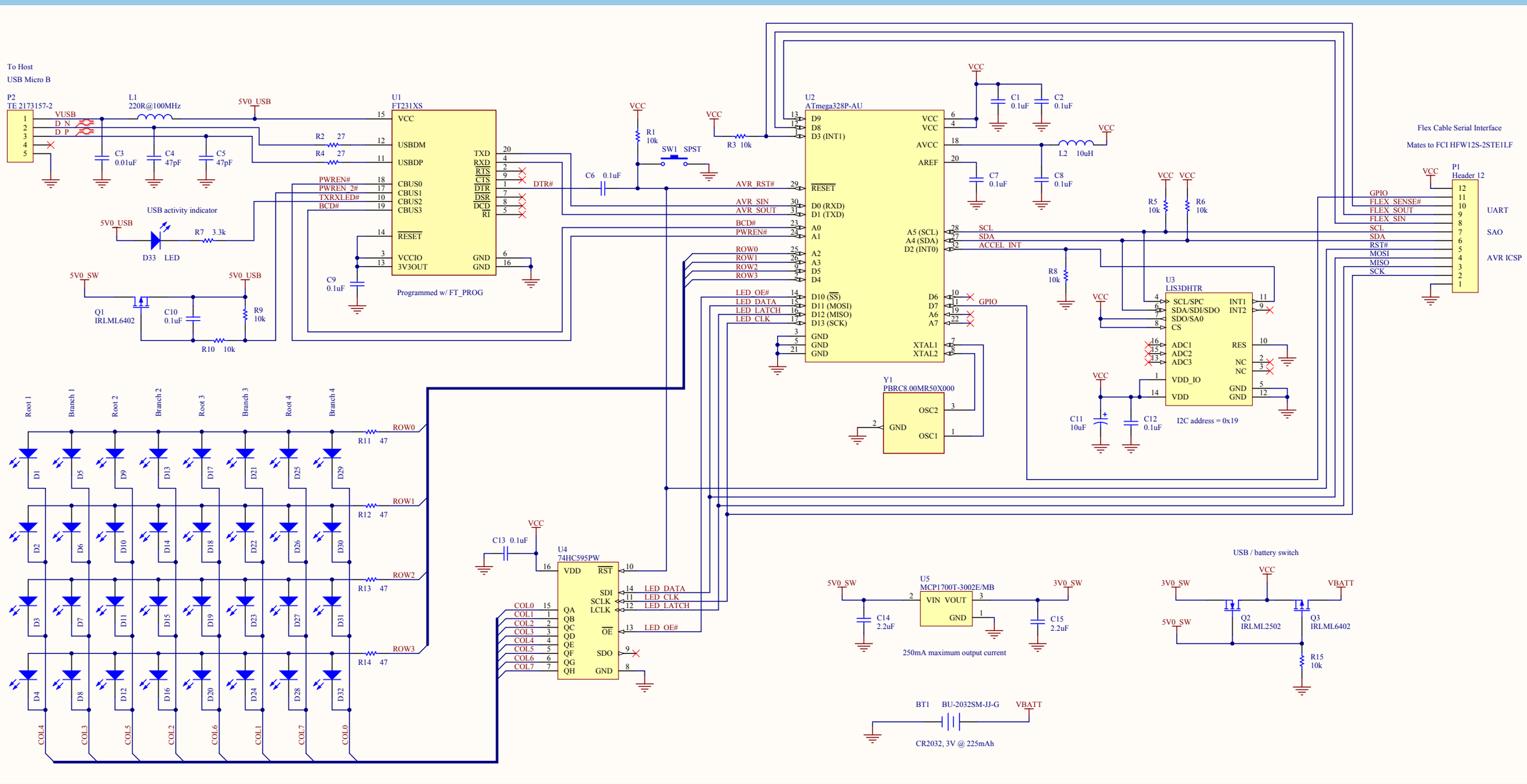
HARDWARE


DEFCON 神州 1.0
CHINA
DEFCON
百度安全
LED
USB
POWER
CONTROL
MCU
ACCELEROMETER
FPC

BLOCK DIAGRAM

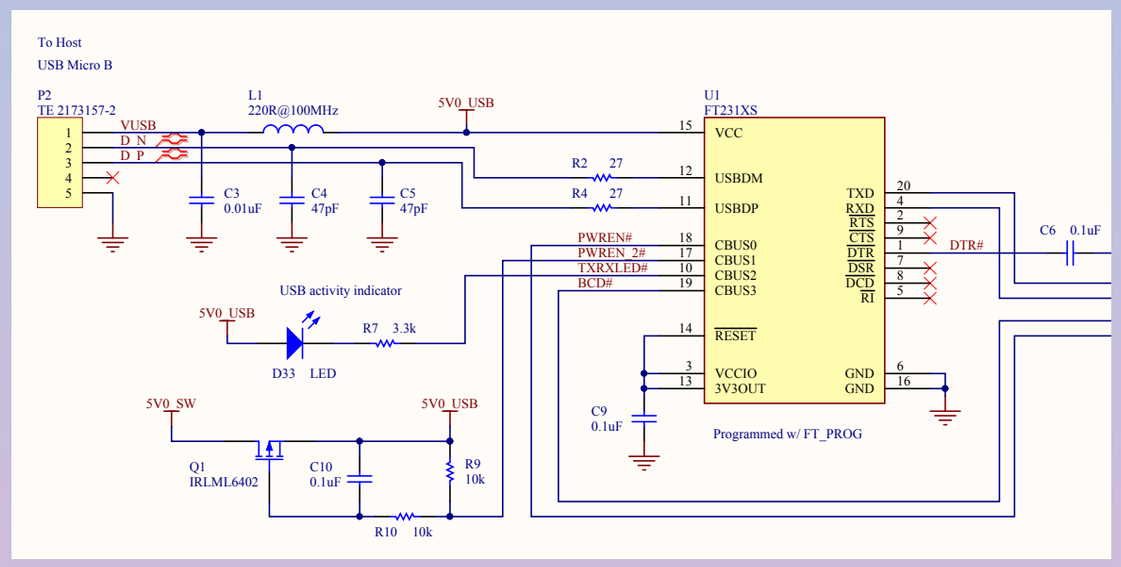


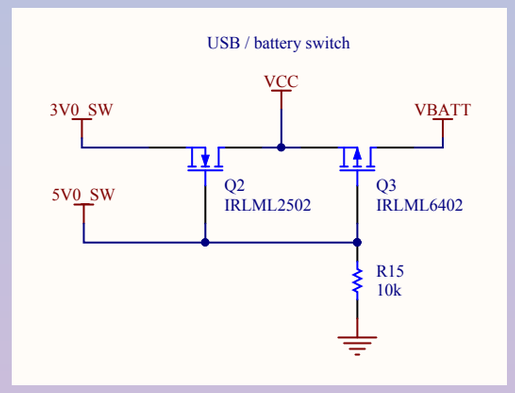
SCHEMATIC

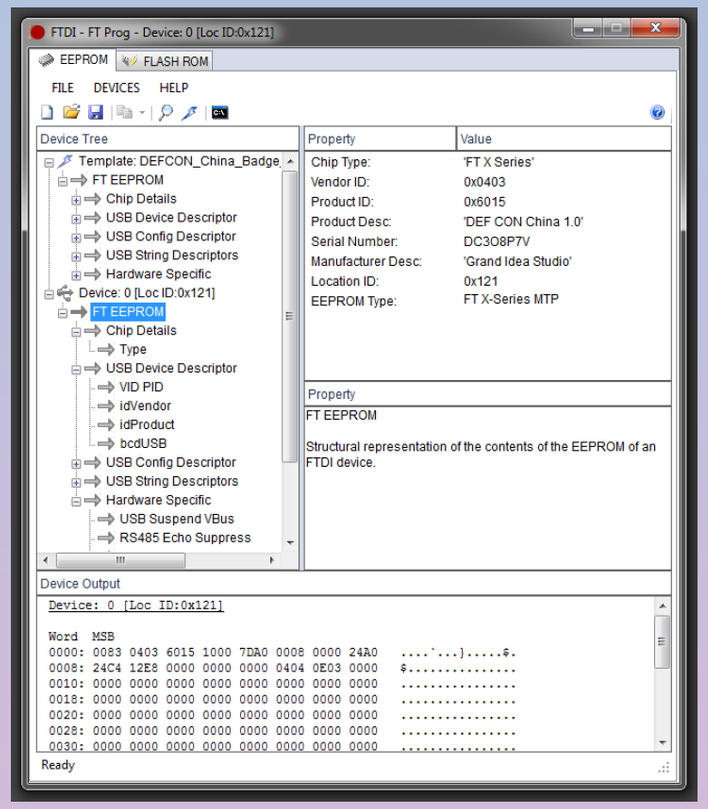

To Host
USB Micro B
P2
TE 2173157-2
VUSB
D_N
D_P
L1
220R@100MHz
5V0_USB
C3 0.01uF
C4 47pF
C5 47pF
U1
FT231XS
VCC
USBDM
USBDP
CBUS0
CBUS1
CBUS2
CBUS3
RESET
VCCIO
3V3OUT
TXD
RXD
RTS
CTS
DTR
DSR
DCD
RI
GND
R2 27
R4 27
PWREN#
PWREN_2#
TXRXLED#
BCD#
DTR#
C6 0.1uF
C9 0.1uF
Programmed w/ FT_PROG
USB activity indicator
D33 LED
R7 3.3k
5V0_SW
Q1
IRLML6402
C10 0.1uF
R9 10k
R10 10k
R1 10k
SW1 SPST
R3 10k
U2
ATmega328P-AU
D9
D8
D3 (INT1)
RESET
D0 (RXD)
D1 (TXD)
A0
A1
A2
A3
D5
D4
D10 (SS)
D11 (MOSI)
D12 (MISO)
D13 (SCK)
GND
GND
GND
VCC
VCC
AVCC
AREF
A5 (SCL)
A4 (SDA)
D2 (INT0)
D6
D7
A6
A7
XTAL1
XTAL2
AVR_RST#
AVR_SIN
AVR_SOUT
BCD#
PWREN#
ROW0
ROW1
ROW2
ROW3
LED_OE#
LED_DATA
LED_LATCH
LED_CLK
GPIO
Y1
PBRC8.00MR50X000
OSC2
GND
OSC1
C1 0.1uF
C2 0.1uF
C7 0.1uF
C8 0.1uF
L2 10uH
R5 10k
R6 10k
R8 10k
SCL
SDA
ACCEL_INT
U3
LIS3DHTR
SCL/SPC
SDA/SDI/SDO
SDO/SA0
CS
ADC1
ADC2
ADC3
VDD_IO
VDD
INT1
INT2
RES
NC
NC
GND
GND
C11 10uF
C12 0.1uF
I2C address = 0x19
Flex Cable Serial Interface
Mates to FCI HFW12S-2STE1LF
P1
Header 12
GPIO
FLEX_SENSE#
FLEX_SOUT
FLEX_SIN
SCL
SDA
RST#
MOSI
MISO
SCK
UART
SAO
AVR ICSP
Root 1
Branch 1
Root 2
Branch 2
Root 3
Branch 3
Root 4
Branch 4
D1
D5
D9
D13
D17
D21
D25
D29
D2
D6
D10
D14
D18
D22
D26
D30
D3
D7
D11
D15
D19
D23
D27
D31
D4
D8
D12
D16
D20
D24
D28
D32
ROW0
ROW1
ROW2
ROW3
R11 47
R12 47
R13 47
R14 47
COL4
COL3
COL5
COL2
COL6
COL1
COL7
COL0
U4
74HC595PW
VDD
RST
SDI
SCLK
LCLK
OE
SDO
GND
LED_DATA
LED_CLK
LED_LATCH
LED_OE#
C13 0.1uF
U5
MCP1700T-3002E/MB
VIN
VOUT
GND
C14 2.2uF
C15 2.2uF
250mA maximum output current
BT1
BU-2032SM-JJ-G
VBATT
CR2032, 3V @ 225mAh
USB / battery switch
3V0_SW
VBATT
Q2
IRLML2502
Q3
IRLML6402
R15 10k

USB INTERFACE

- ALLOWS FOR ARDUINO PROGRAMMING AND INTERACTIVE MODE
- FT231X USB-TO-SERIAL UART
  - ENTIRE USB PROTOCOL HANDLED ON-CHIP
  - HOST WILL RECOGNIZE AS VIRTUAL SERIAL DEVICE/COM PORT
- MOSFETS FOR SOFT-START AND POWER SWITCHOVER

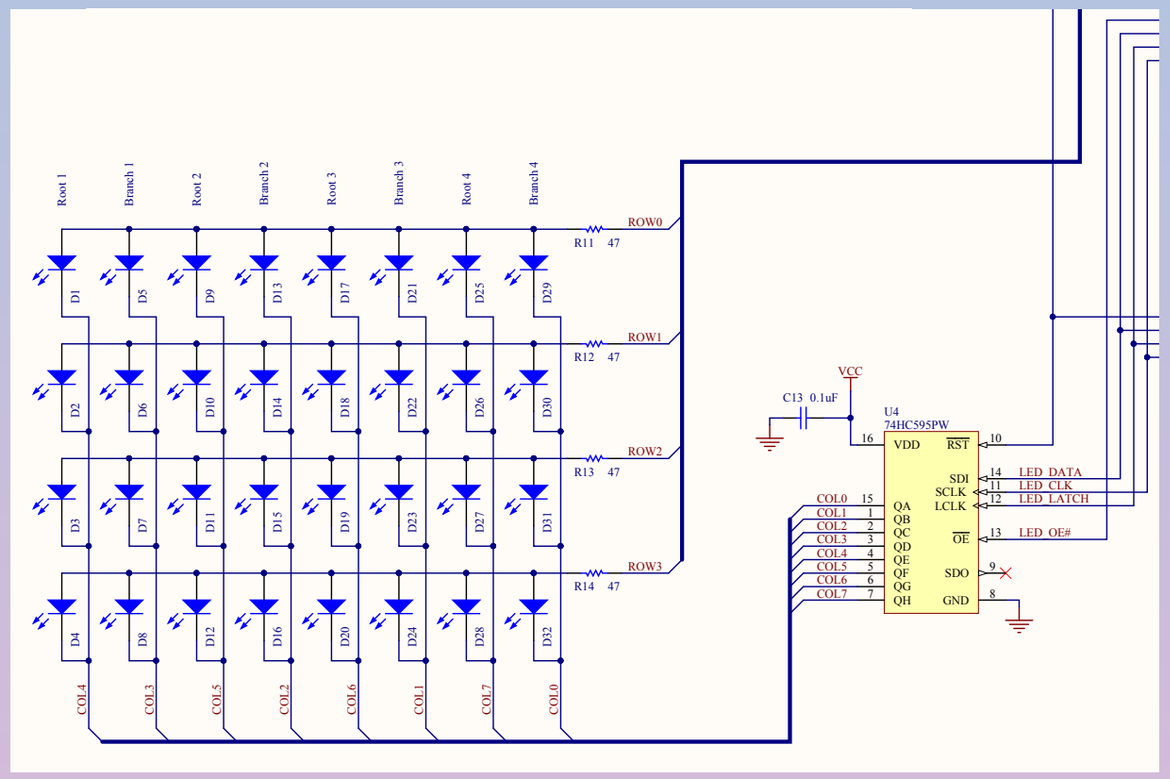


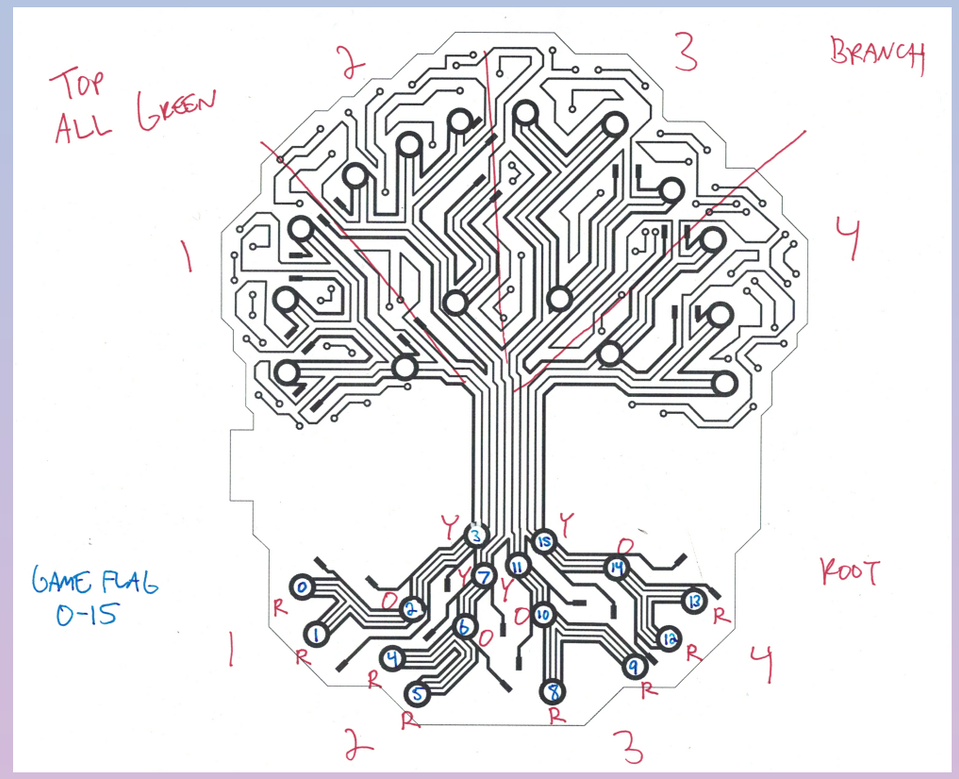




LED MATRIX

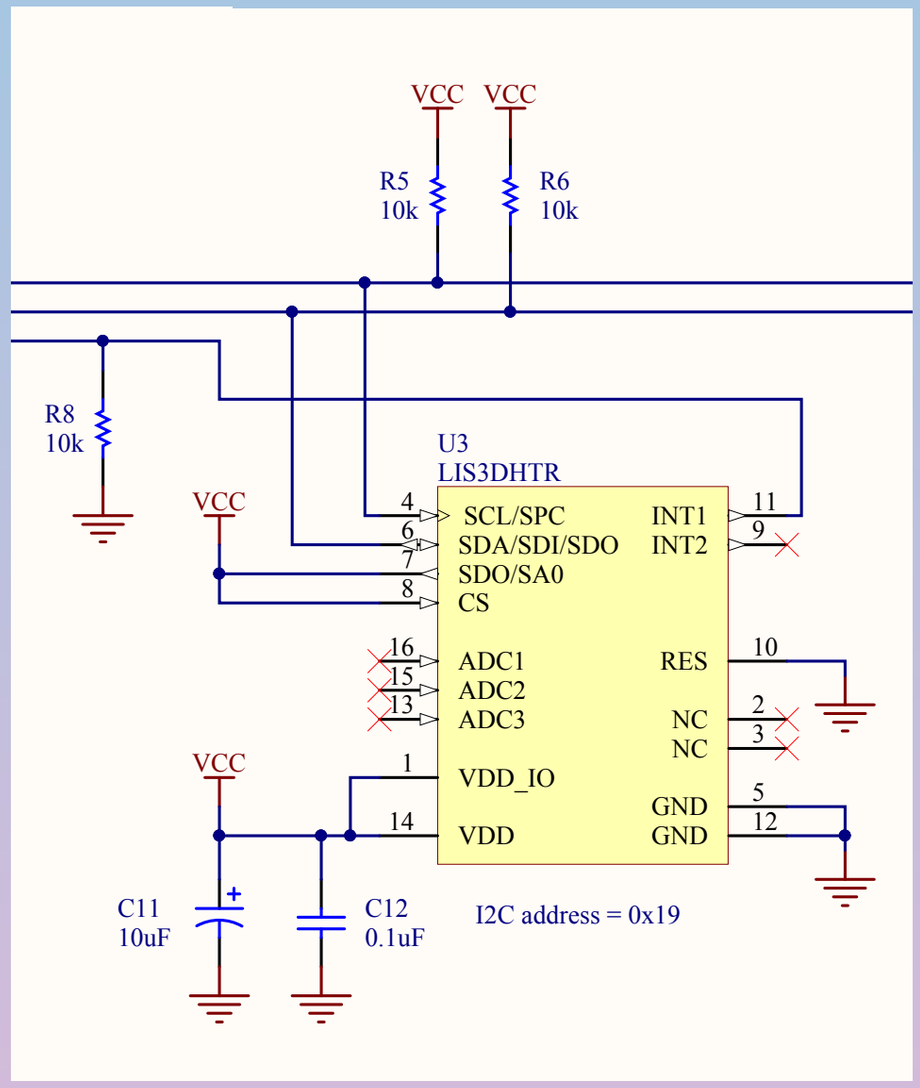
- MULTIPLEXING VIA LED MATRIX LIBRARY
  - ROW CONTROLLED BY DISCRETE I/O
  - COLUMN CONTROLLED THROUGH 74HC595 SHIFT REGISTER
  - REFRESH @ 175HZ TO REDUCE FLICKER
- EACH LED INDIVIDUALLY ADDRESSABLE, DIMMABLE (16 LEVELS)





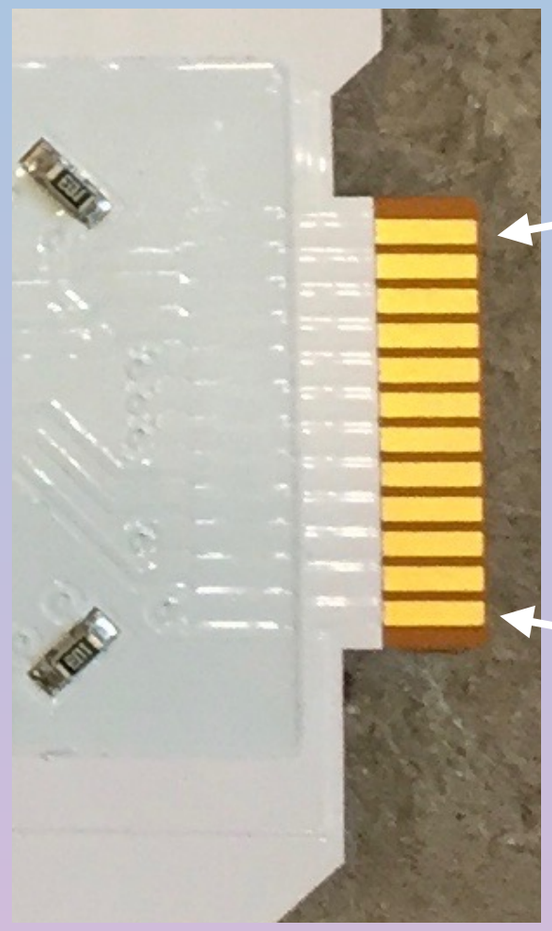
ACCELEROMETER

- ST MICROELECTRONICS LIS3DH
  - 3-AXIS DIGITAL OUTPUT (I2C/SPI)
  - +/- 2, 4, 8, 16G RANGE
  - INTERRUPT ON MOTION OR FREE FALL
- USED TO PRESERVE BATTERY LIFE
  - SLEEP MODE @ 10 SECONDS OF INACTIVITY
- RAW VALUES AVAILABLE THROUGH INTERACTIVE MODE



FLEXIBLE PRINTED CIRCUIT (FPC)

- EDGE CONNECTOR AS INTERFACE TO THE OUTSIDE WORLD
  - UART, I2C, AVR ICSP
- USED WITH PROGRAMMING SHIELD TO SET/READ STATE OF BADGE LEDS



1. GND	7. SCL
2. SCK	8. SIN
3. MISO	9. SOUT
4. MOSI	10. /SENSE
5. /RST	11. GPIO
6. SDA	12. VCC

BILL-OF-MATERIALS

Item	Quantity	Reference	Manufacturer	Manuf. Part #	Distributor	Distrib. Part #	Description
1	1	BT1	MPD	BU2032SM-JJ-GTR	N/A	N/A	Battery Holder, CR2032, SMD
1b	1	N/A	Panasonic	CR2032	Digi-Key	P189-ND	Battery, Coin Cell, Lithium, 3V, 225mAh
2	9	C1, C2, C6, C7, C8, C9, C10, C12, C13	Kemet	C0603C104K4RACTU	Digi-Key	399-1096-2-ND	Capacitor, 0.1uF, 16V, Ceramic, 10%, X7R, 0603
3	1	C3	Samsung	CL10B103KB8NCNC	Digi-Key	1276-1921-2-ND	Capacitor, 0.01uF, 50V, Ceramic, 10%, X7R, 0603
4	2	C4, C5	AVX	06035A470JAT2A	Digi-Key	478-1171-2-ND	Capacitor, 47pF, 50V, Ceramic, 5%, C0G/NP0, 0603
5	1	C11	Kemet	T491A106M016AT	Digi-Key	399-3687-2-ND	Capacitor, 10uF, 16V, Tantalum, 20%, Size A
6	2	C14, C15	Taiyo Yuden	TMK212B7225KG-TR	Digi-Key	587-2991-2-ND	Capacitor, 2.2uF, 25V, Ceramic, 10%, X7R, 0805
7	4	D1, D9, D17, D25	Kingbright	AA3528SYCKT09	N/A	N/A	LED, Yellow, 250mcd, 2.0Vf, 590nm, Reverse Mount, PLCC-2
8	4	D2, D10, D18, D26	Kingbright	AA3528SECKT09	N/A	N/A	LED, Orange, 350mcd, 2.1Vf, 605nm, Reverse Mount, PLCC-2
9	8	D3, D4, D11, D12, D19, D20, D27, D28	Kingbright	AA3528SURCKT09	N/A	N/A	LED, Red, 350mcd, 1.95Vf, 630nm, Reverse Mount, PLCC-2
10	16	D5, D6, D7, D8, D13, D14, D15, D16, D21, D22, D23, D24, D29, D30, D31, D32	Kingbright	AA3528CGCKT09	N/A	N/A	LED, Green, 100mcd, 2.1Vf, 570nm, Reverse Mount, PLCC-2
11	1	D33	Kingbright	APT1608LQWF/D	N/A	N/A	LED, White, 35mcd, 2.65Vf, 0603
12	1	L1	TDK	MPZ2012S221AT000	Digi-Key	445-1568-2-ND	Inductor, Ferrite Bead, 220R @ 100MHz, 3A, 0805
13	1	L2	Taiyo Yuden	LBMF1608T100K	Digi-Key	587-1714-2-ND	Inductor, Wirewound, 10uH, 10%, 360mR, 80mA, 0603
14	1	P2	TE Connectivity	2173157-2	Verical	N/A	Connector, Micro-USB Type B, R/A, 5 position, SMD
15	2	Q1, Q3	Infineon	IRLML6402TRPBF	Digi-Key	IRLML6402PBFTR-ND	Transistor, MOSFET, P-Channel, 20V, 65mR @ 3.7A, SOT23
16	1	Q2	Infineon	IRLML2502TRPBF	Digi-Key	IRLML2502TRPBFTR-ND	Transistor, MOSFET, N-Channel, 20V, 45mR @ 4.2A, SOT23
17	8	R1, R3, R5, R6, R8, R9, R10, R15	Panasonic	ERJ-3GEYJ103V	Digi-Key	P10KGTR-ND	Resistor, 10k, 5%, 1/10W, 0603
18	2	R2, R4	Panasonic	ERJ-3GEYJ270V	Digi-Key	P27GTR-ND	Resistor, 27 ohm, 5%, 1/10W, 0603
19	1	R7	Panasonic	ERJ-3GEYJ332V	Digi-Key	P3.3KGTR-ND	Resistor, 3.3k, 5%, 1/10W, 0603
20	4	R11, R12, R13, R14	Panasonic	ERJ-3GEYJ470V	Digi-Key	P47GTR-ND	Resistor, 47 ohm, 5%, 1/10W, 0603
21	1	SW1	Panasonic	EVP-AA202K	Digi-Key	P13348SDKR-ND	Switch, SPST, Tactile Momentary, 160gf, 3.5 x 2.9mm, J-Lead
22	1	U1	FTDI	FT231XS-R	Digi-Key	768-1129-2-ND	IC, USB-to-UART Bridge, SSOP20
23	1	U2	Microchip	ATMEGA328P-AU	N/A	N/A	IC, Microcontroller, 32KB Flash, TQFP32
23b	1	N/A	Microchip	N/A	N/A	N/A	IC, Microcontroller, Programming
24	1	U3	STMicroelectronics	LIS3DHTR	Digi-Key	497-10613-6-ND	IC, Accelerometer, 3-Axis, 2-16g, LGA16
25	1	U4	Nexperia	74HC595PW,118	Digi-Key	1727-3068-2-ND	IC, Shift Register, 8-bit, TSSOP16
26	1	U5	Microchip	MCP1700T-3002E/MB	Digi-Key	MCP1700T-3002E/MBCT-ND	Voltage Regulator, LDO, 3.0V, 250mA, SOT89-3
27	1	Y1	Kyocera	PBRC8.00MR50X000	Mouser	581-PBRC8.00MR50X	Resonator, 8MHz, 0.5%, Internal 15pF Capacitor, SMD
28	1	PCB	Electronic Interconnect	DCN1.0	N/A	N/A	PCB Fabrication, Assembly, Test

FIRMWARE

- ARDUINO
  - OPEN SOURCE PLATFORM BASED ON EASY-TO-USE HW/SW/FW
  - WORLDWIDE COMMUNITY OF USERS/CONTRIBUTORS
- 90% OF FLASH (27.6KB), 43% OF RAM (887 BYTES)
- LOOP
  - SET POWER STATE (BATTERY, USB, USB CHARGER)
  - CHECK FOR/PROCESS INTERACTIVE MODE
  - CHECK FOR/PROCESS FPC COMMUNICATION
  - UPDATE LEDS
  - SLEEP UNTIL ACCELEROMETER INTERRUPT

ARDUINO CHEAT SHEET

Structure & Flow

Basic Program Structure
```
void setup() {
  // Runs once when sketch starts
}
void loop() {
  // Runs repeatedly
}
```

Control Structures
```
if (x < 5) { ... } else { ... }
while (x < 5) { ... }
for (int i = 0; i < 10; i++) { ... }
break;     // Exit a loop immediately
continue;  // Go to next iteration
switch (var) {
  case 1:
    ...
    break;
  case 2:
    ...
    break;
  default:
    ...
}
return x;  // x must match return type
return;    // For void return type
```

Function Definitions
```
<ret. type> <name>(<params>) { ... }
e.g. int double(int x) {return x*2;}
```

Operators

General Operators

=	assignment		
+	add	-	subtract
*	multiply	/	divide
%	modulo		
==	equal to	!=	not equal to
<	less than	>	greater than
<=	less than or equal to		
>=	greater than or equal to		
&&	and	||	or
!	not		

Compound Operators

++	increment
--	decrement
+=	compound addition
-=	compound subtraction
*=	compound multiplication
/=	compound division
&=	compound bitwise and
|=	compound bitwise or

Bitwise Operators

&	bitwise and	|	bitwise or
^	bitwise xor	~	bitwise not
<<	shift left	>>	shift right

Pointer Access

&	reference: get a pointer
*	dereference: follow a pointer

Built-in Functions

Pin Input/Output
```
Digital I/O - pins 0-13 A0-A5
  pinMode(pin,
    [INPUT, OUTPUT, INPUT_PULLUP])
  int digitalRead(pin)
  digitalWrite(pin, [HIGH, LOW])

Analog In - pins A0-A5
  int analogRead(pin)
  analogReference(
    [DEFAULT, INTERNAL, EXTERNAL])

PWM Out - pins 3 5 6 9 10 11
  analogWrite(pin, value)
```

Advanced I/O
```
tone(pin, freq_Hz)
tone(pin, freq_Hz, duration_ms)
noTone(pin)
shiftOut(dataPin, clockPin,
  [MSBFIRST, LSBFIRST], value)
unsigned long pulseIn(pin,
  [HIGH, LOW])
```

Time
```
unsigned long millis()
  // Overflows at 50 days
unsigned long micros()
  // Overflows at 70 minutes
delay(msec)
delayMicroseconds(usec)
```

Math
```
min(x, y)   max(x, y)   abs(x)
sin(rad)    cos(rad)    tan(rad)
sqrt(x)     pow(base, exponent)
constrain(x, minval, maxval)
map(val, fromL, fromH, toL, toH)
```

Random Numbers
```
randomSeed(seed) // long or int
long random(max) // 0 to max-1
long random(min, max)
```

Bits and Bytes
```
lowByte(x)    highByte(x)
bitRead(x, bitn)
bitWrite(x, bitn, bit)
bitSet(x, bitn)
bitClear(x, bitn)
bit(bitn)  // bitn: 0=LSB 7=MSB
```

Type Conversions
```
char(val)       byte(val)
int(val)        word(val)
long(val)       float(val)
```

External Interrupts
```
attachInterrupt(interrupt, func,
 [LOW, CHANGE, RISING, FALLING])
detachInterrupt(interrupt)
interrupts()
noInterrupts()
```

Libraries

Serial - comm. with PC or via RX/TX
```
begin(long speed)  // Up to 115200
end()
int available()  // #bytes available
int read()   // -1 if none available
int peek()   // Read w/o removing
flush()
print(data)    println(data)
write(byte)    write(char * string)
write(byte * data, size)
SerialEvent()  // Called if data rdy
```

SoftwareSerial.h - comm. on any pin
```
SoftwareSerial(rxPin, txPin)
begin(long speed)  // Up to 115200
listen()       // Only 1 can listen
isListening()  // at a time.
read, peek, print, println, write
  // Equivalent to Serial library
```

EEPROM.h - access non-volatile memory
```
byte read(addr)
write(addr, byte)
EEPROM[index]  // Access as array
```

Servo.h - control servo motors
```
attach(pin, [min_uS, max_uS])
write(angle)  // 0 to 180
writeMicroseconds(uS)
   // 1000-2000; 1500 is midpoint
int read()    // 0 to 180
bool attached()
detach()
```

Wire.h - I²C communication
```
begin()     // Join a master
begin(addr) // Join a slave @ addr
requestFrom(address, count)
beginTransmission(addr) // Step 1
send(byte)              // Step 2
send(char * string)
send(byte * data, size)
endTransmission()       // Step 3
int available() // #bytes available
byte receive()  // Get next byte
onReceive(handler)
onRequest(handler)
```

Variables, Arrays, and Data

Data Types

boolean	true | false
char	-128 - 127, 'a' '$' etc.
unsigned char	0 - 255
byte	0 - 255
int	-32768 - 32767
unsigned int	0 - 65535
word	0 - 65535
long	-2147483648 - 2147483647
unsigned long	0 - 4294967295
float	-3.4028e+38 - 3.4028e+38
double	currently same as float
void	i.e., no return value

Strings
```
char str1[8] =
  {'A','r','d','u','i','n','o','\0'};
  // Includes \0 null termination
char str2[8] =
  {'A','r','d','u','i','n','o'};
  // Compiler adds null termination
char str3[] = "Arduino";
char str4[8] = "Arduino";
```

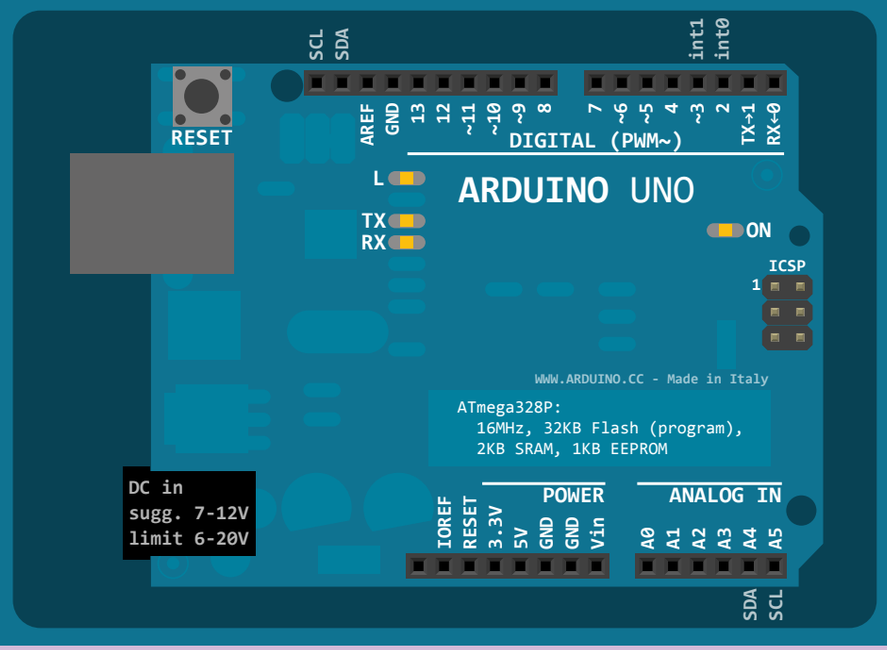
Numeric Constants

123	decimal
0b01111011	binary
0173	octal - base 8
0x7B	hexadecimal - base 16
123U	force unsigned
123L	force long
123UL	force unsigned long
123.0	force floating point
1.23e6	1.23*10^6 = 1230000

Qualifiers

static	persists between calls
volatile	in RAM (nice for ISR)
const	read-only
PROGMEM	in flash

Arrays
```
int myPins[] = {2, 4, 8, 3, 6};
int myInts[6];   // Array of 6 ints
myInts[0] = 42;  // Assigning first
                 // index of myInts
myInts[6] = 12;  // ERROR! Indexes
                 // are 0 though 5
```



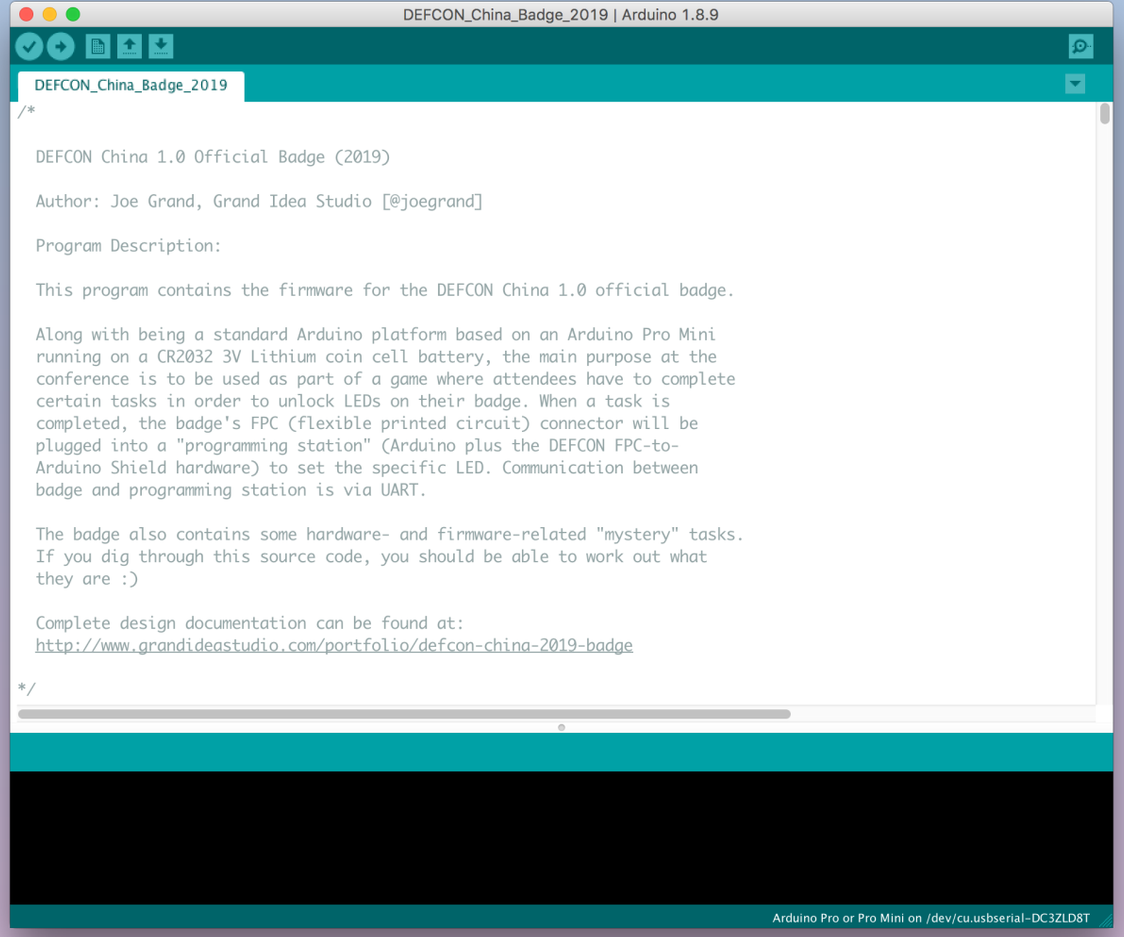
by Mark Liffiton
version: 2018-08-06

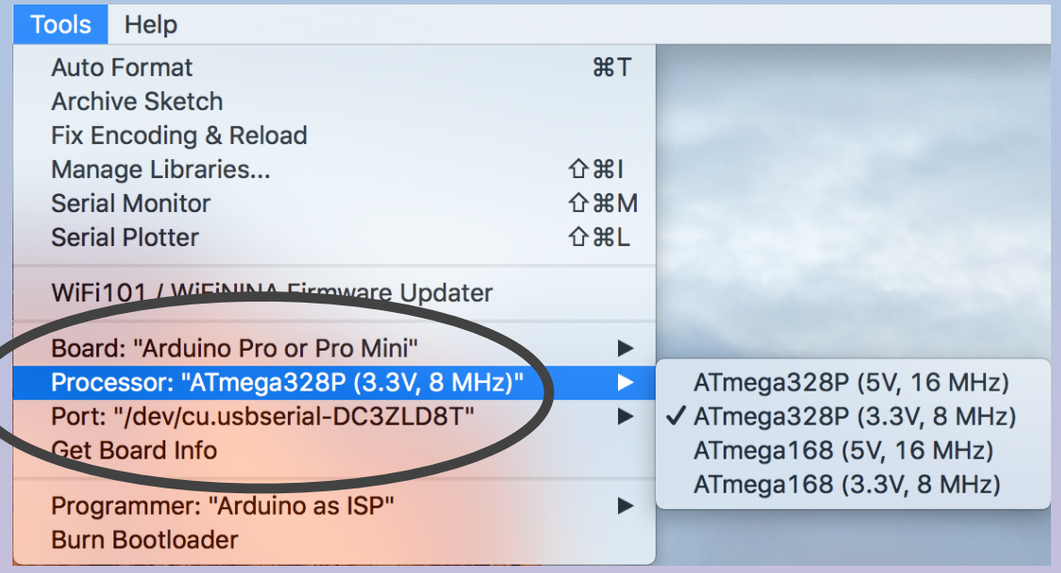
source: https://github.com/liffiton/Arduino-Cheat-Sheet/

Adapted from:
- Original: Gavin Smith
- SVG version: Frederic Dufourg
- Arduino board drawing: Fritzing.org

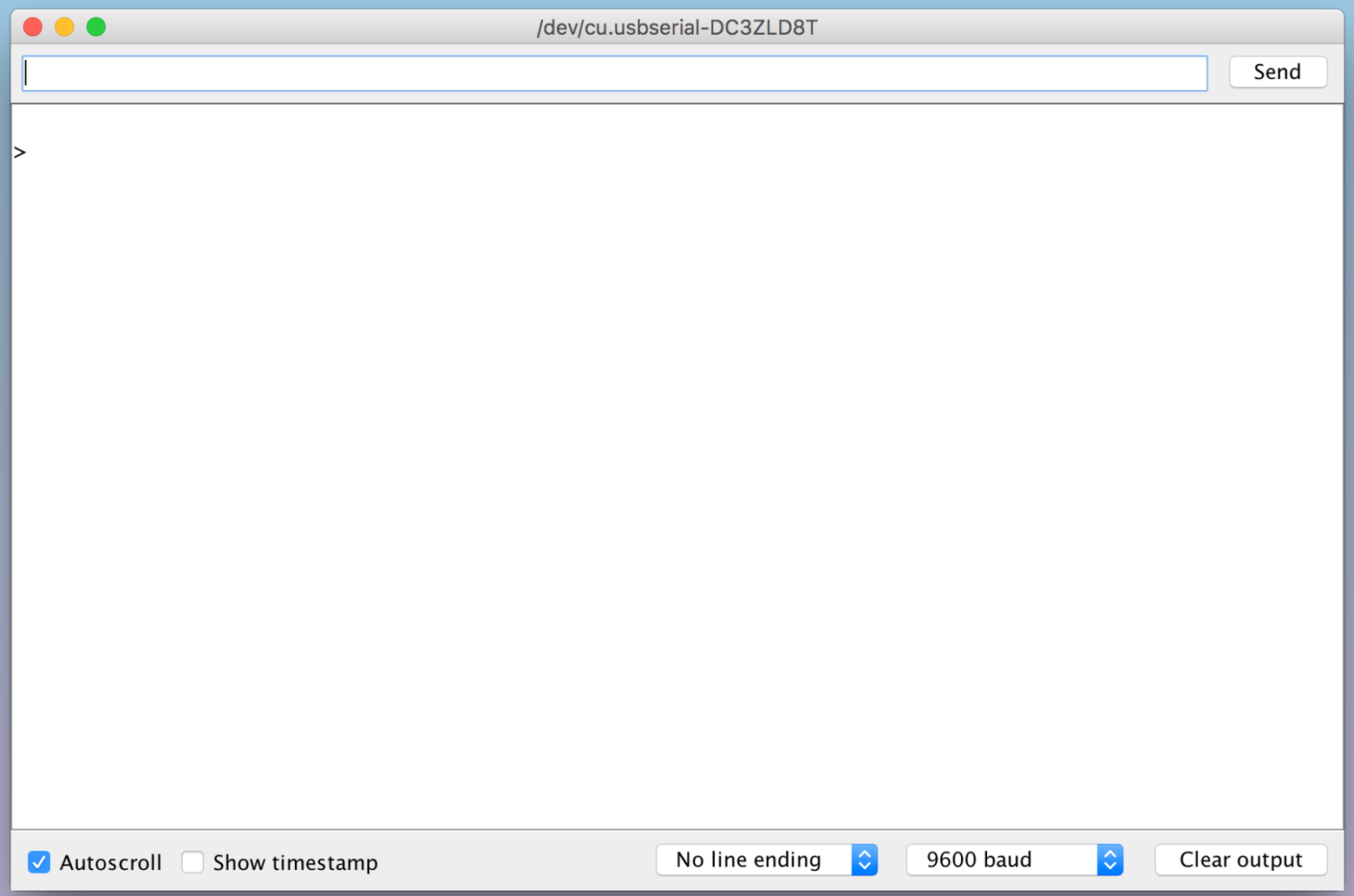
SETUP DEVELOPMENT ENVIRONMENT

- ARDUINO IDE
  - CROSS PLATFORM (WINDOWS, MAC OS, LINUX)
  - WRITTEN IN JAVA, BASED ON PROCESSING
  - `www.arduino.cc/en/Main/Software`





INTERACT W/ BADGE VIA SERIAL MONITOR



LIBRARIES

- THIRD-PARTY LIBRARIES TO ADD FUNCTIONALITY TO ARDUINO
  - ESSENTIAL FOR RAPID DEVELOPMENT
  - SOME CODE MODIFICATIONS REQUIRED DURING BADGE INTEGRATION
- LOW POWER
  - https://github.com/rocketscream/Low-Power
- ADAFRUIT_LIS3DH (ACCELEROMETER)
  - https://github.com/adafruit/Adafruit_LIS3DH
- ADAFRUIT_SENSOR (SENSOR ABSTRACTION LAYER)
  - https://github.com/adafruit/Adafruit_Sensor

LIBRARIES

- LED MATRIX (INDIVIDUALLY ADDRESSABLE, DIMMABLE, SHIFT REGISTER)
  - https://github.com/marcmerlin/LED-Matrix
- ADAFRUIT-GFX-LIBRARY (CORE GRAPHICS PRIMITIVES)
  - https://github.com/adafruit/Adafruit-GFX-Library
- DIO2 (FAST DIGITAL I/O)
  - www.codeproject.com/Articles/732646/Fast-digital-I-O-for-Arduino
- TIMERONE (ENHANCED TIMER, PERIODIC INTERRUPTS)
  - https://github.com/PaulStoffregen/TimerOne

INSTALL LIBRARIES

CODE MODIFICATIONS

- LED MATRIX
  - ADD #DEFINE SWAP() TO .CPP TO PREVENT COMPILING ERROR
  - REMOVE #DEFINES FOR DIO2 PINMODE AND DIGITALWRITE
    - CONFLICTED WITH MY CORE CODE

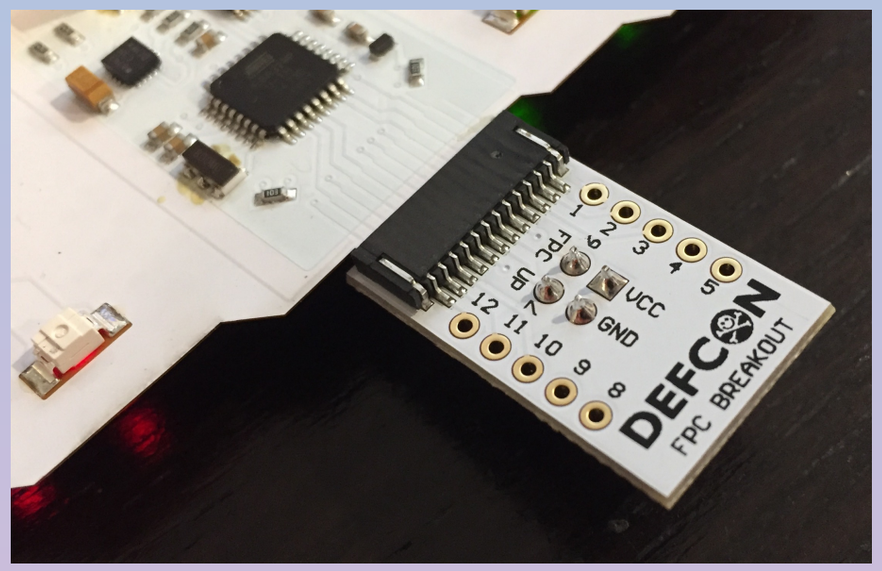
EXPLORE SOURCE CODE

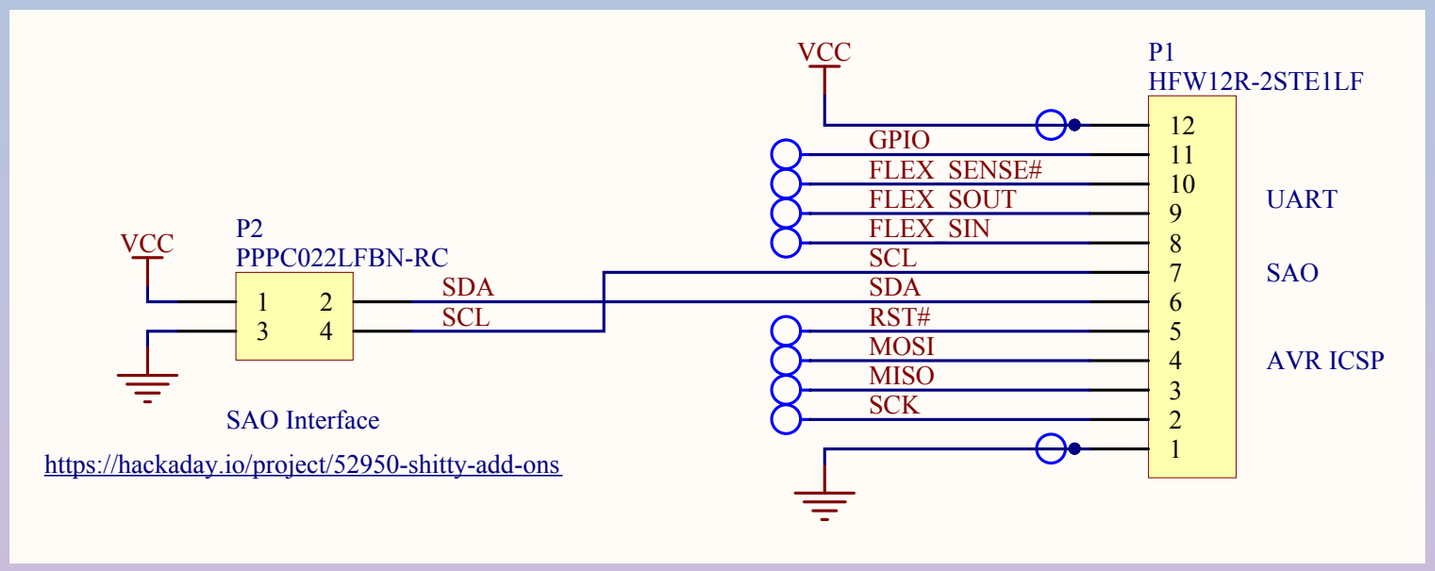
- FIND FLAGS AND FIGURE OUT HOW TO ACHIEVE THEM
  - ENABLE SPECIAL BADGE HACKING WORKSHOP FLAG
  - ???

COMPILE & UPLOAD CODE

FPC BREAKOUT BOARD

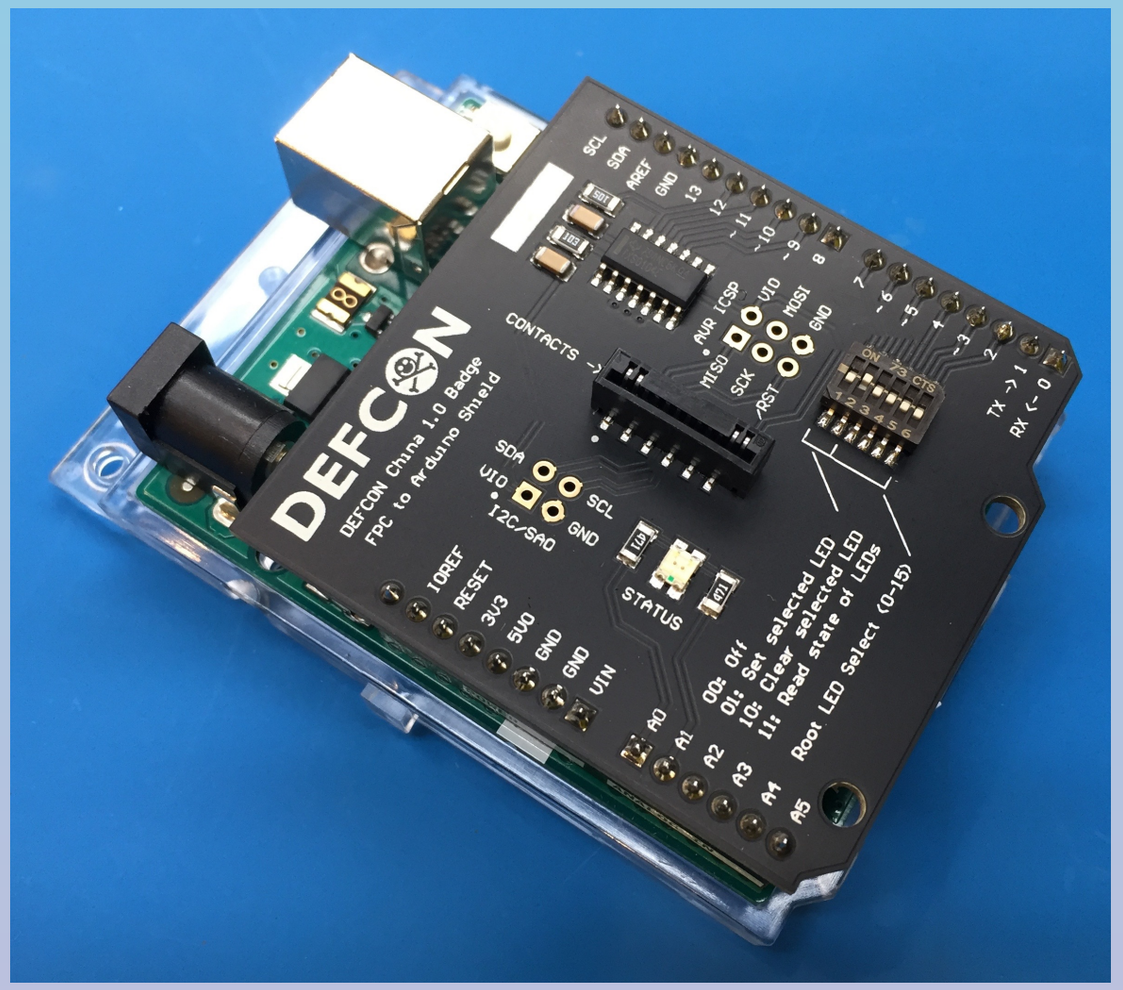
- ACCESS ALL FPC SIGNALS
  - UART, I2C, AVR ICSP
- SAO ADAPTER
- `http://oshpark.com/shared_projects/X4QDh3nj`



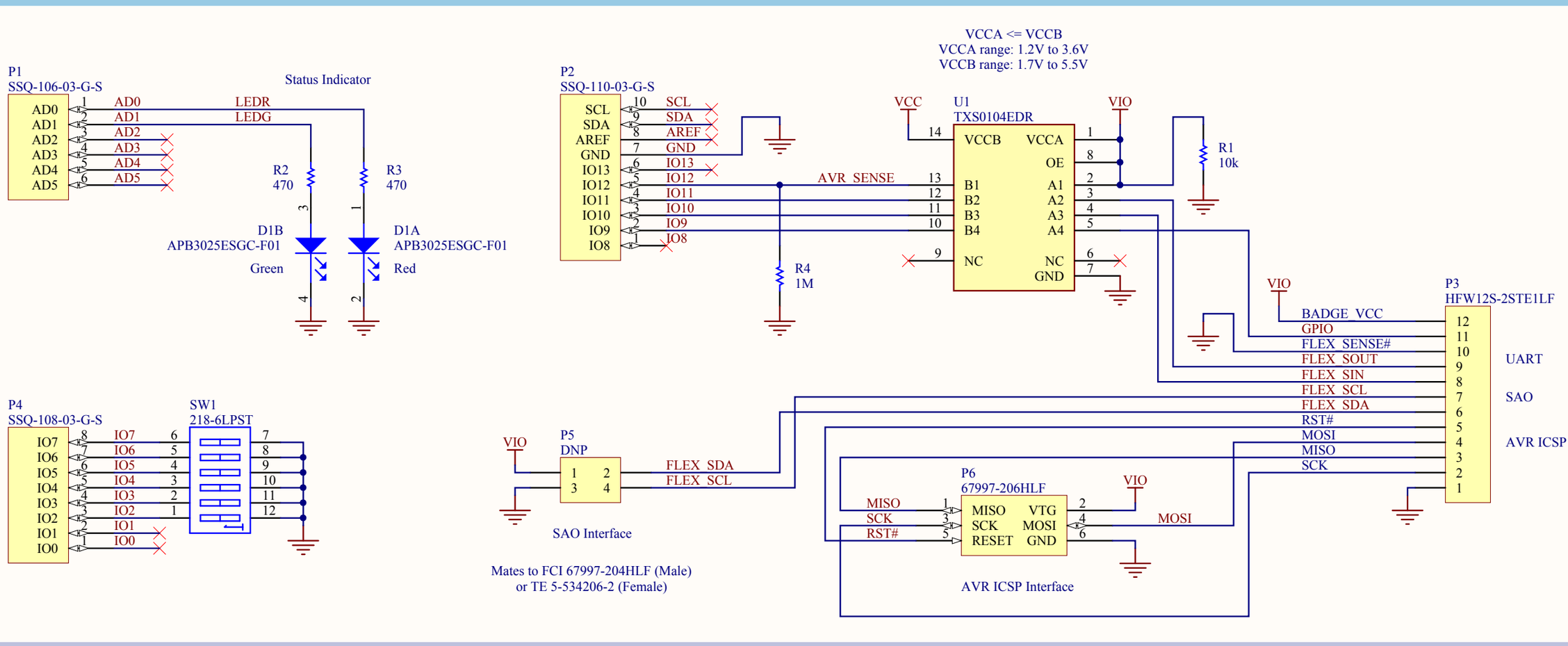


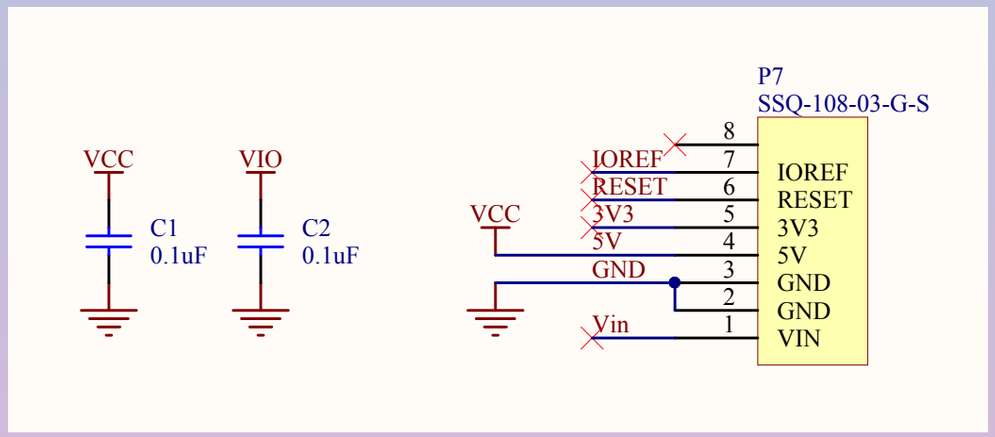
PROGRAMMING SHIELD

- SERIAL COMMUNICATION VIA FPC
- SET/CLEAR INDIVIDUAL LED
- READ STATE OF BADGE
- ARDUINO W/ CUSTOM SHIELD
  - TXS0104 LEVEL TRANSLATOR
    - 5V ARDUINO <-> 3V BADGE
  - DIP SWITCHES
  - I2C, AVR ICSP FOOTPRINTS
- http://oshpark.com/shared_projects/WGHZCah0



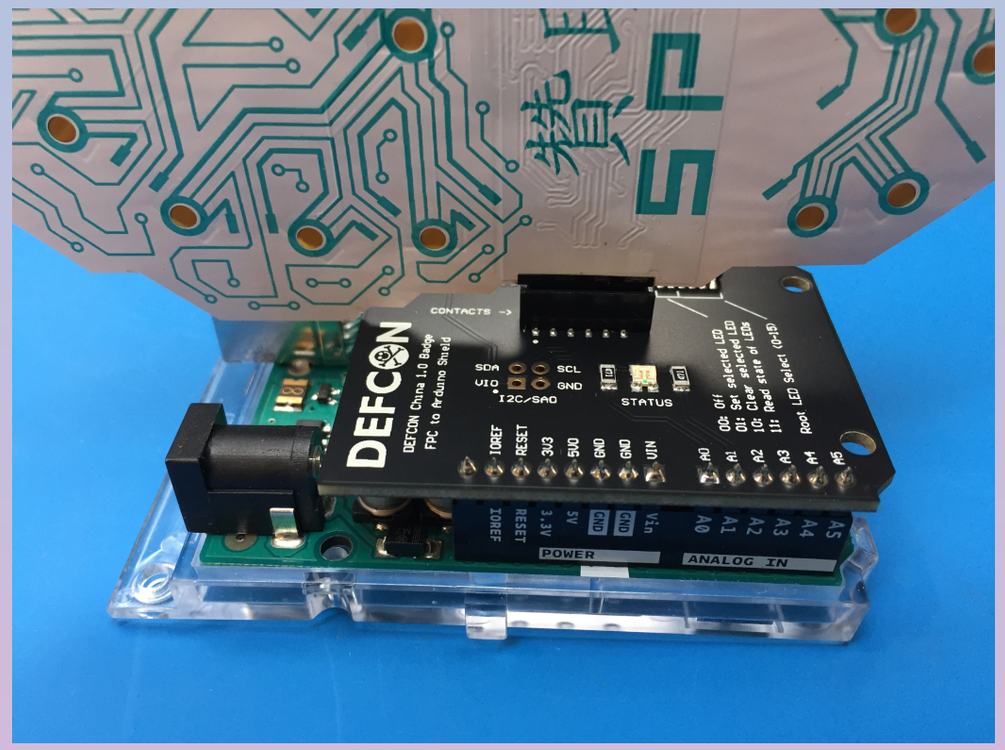
PROGRAMMING SHIELD

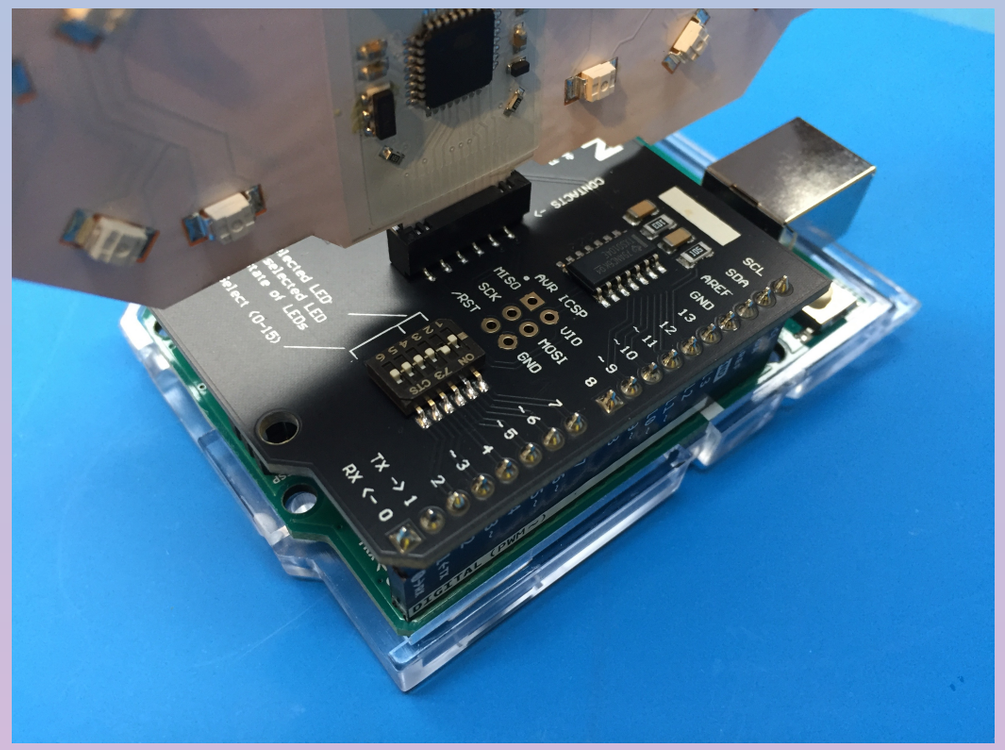




PROGRAMMING SHIELD

- DIP SWITCH SETTINGS DETERMINE FUNCTIONALITY
  - 00: OFF
  - 01: SET SELECTED LED
  - 10: CLEAR SELECTED LED
  - 11: READ BADGE STATE





HACKING

- FLEXIBLE, GENERAL PURPOSE ARDUINO PLATFORM
  - ISOLATE CORE HARDWARE FROM ROOTS/BRANCHES (TREE TRIMMING)?
  - MODIFY FW FOR BETTER LED ANIMATIONS?
  - MORE INTERACTION W/ ACCELEROMETER?
  - ???
- DESIGN DOCUMENTATION, CODE, ETC.
  - `www.grandideastudio.com/portfolio/defcon-china-2019-badge`

OPEN LAB

THANK YOU FOR COMING!

@JOEGRAND | WWW.GRANDIDEASTUDIO.COM