UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Sensory Awareness Device for Bars and Restaurants

Megan Heinhold, Evan Lindquist, Carl Wolff

Group 17 Final Presentation

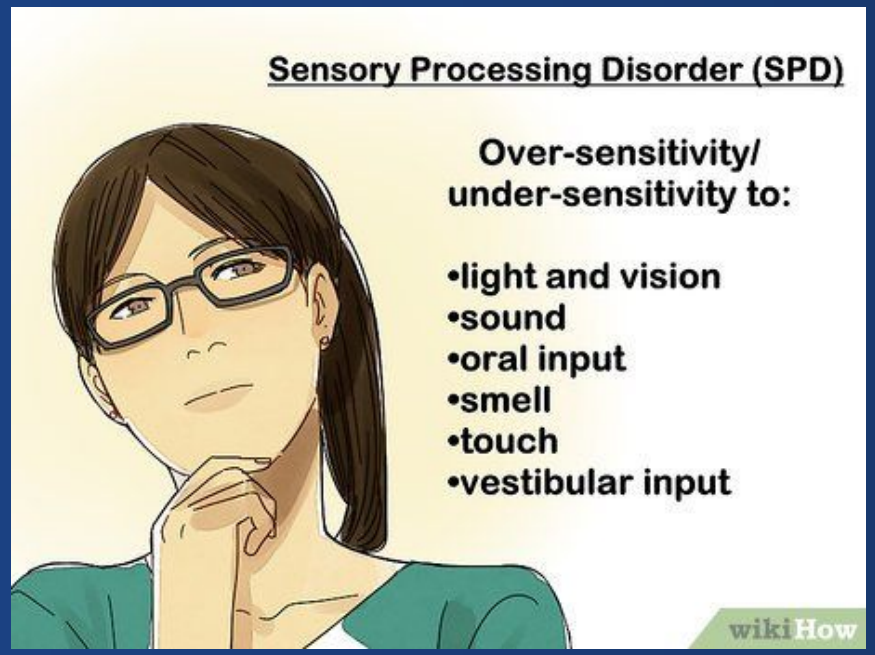
April 29, 2022

Introduction

Our personal experiences involving bars, restaurants, and the environments they create.

Objective

Create a device for bar and restaurant owners that would allow their businesses to be more accessible to those with sensory processing disorders.



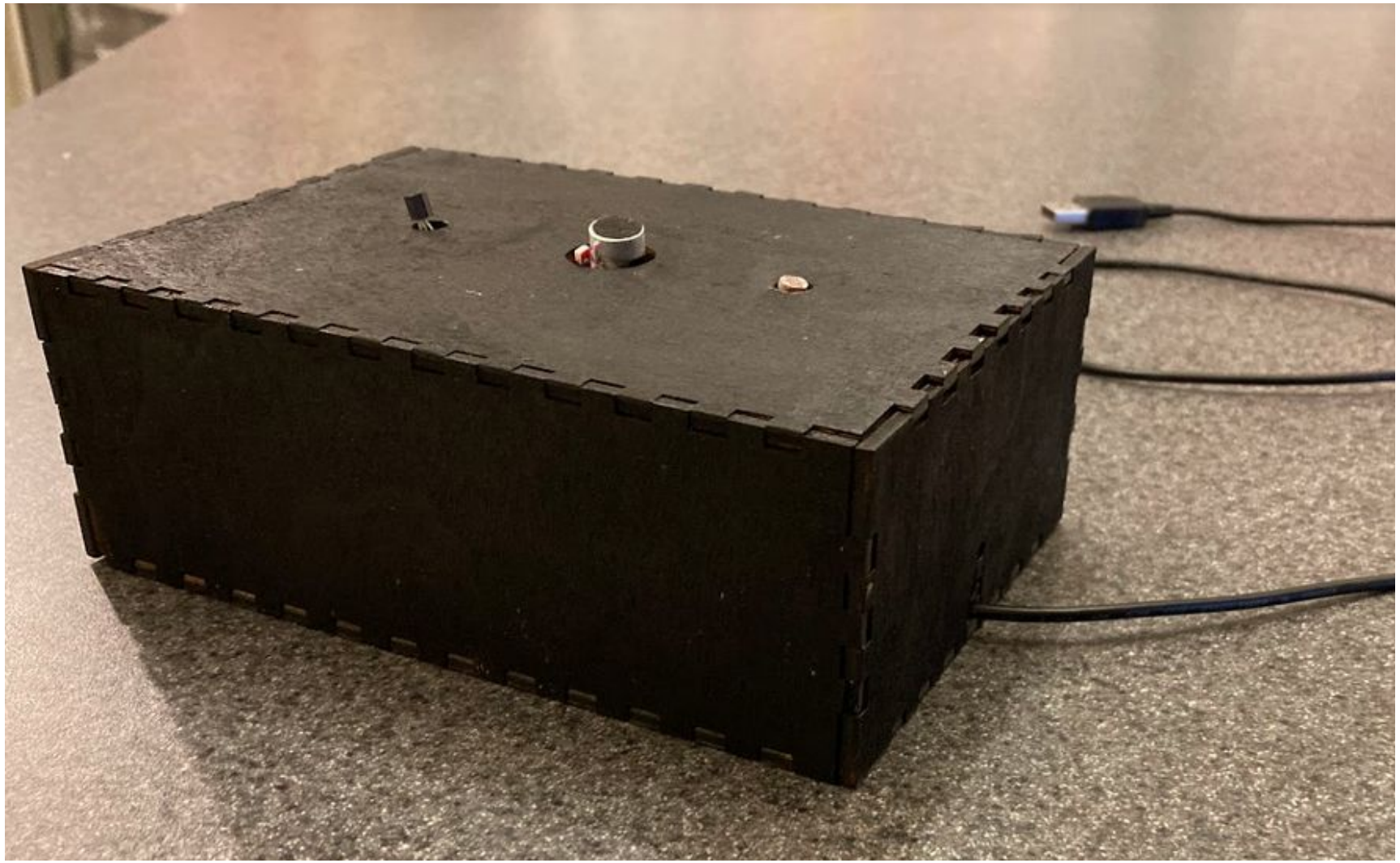
Our entire product must be simple, small, and effective.

In order for business owners to elect for the use of our device, certain parameters must be met. The device must be simple to set up/use, unobtrusive, and useful in the information it presents.

In order for bar and restaurant patrons to want to use our data, it must be accessible and easily understood, such that those who want to view the information have minimal difficulty doing so.

Our Solution

Final Product - Physical Unit



Final Product - Web Application

View Data | About the Project

Sensory Awareness, Champaign-Urbana

Project for SP22 ECE 445 by Megan Heinhold, Evan Lindquist, and Carl Wolff

About the data

Flashing lights are defined as light levels changing drastically at a rate of **10-12 Hz**.

Dangerous sound levels refer to noise that peaks above **110 dB**.

Each of these circumstances pose a threat to your health and safety, especially if you have epilepsy, a sensory processing disorder, or another condition that might make you more bothered by sensory overload.

Overall sound falls into **"little background noise," "moderate background noise," "significant background noise or light music," "significant background noise or loud music," and "very loud"** categories.

Overall light falls into **"very dark," "dark," "ambient lighting", "bright," and "very bright"**.

Device One

Last updated: **April 27, 2022 at 02:18PM**

Dangerous sound levels detected in last 10 minutes? **yes**

Flashing lights detected in last 10 minutes? **no**

Sound level: **moderate background noise**

Light level: **very bright**

Temperature: **73.84 F**

Murphy's

Dangerous sound levels detected in last 10 minutes? yes

Flashing lights detected in last 10 minutes? no

Ambient sound level: loud

Ambient light level: dark

Temperature: 85 F

Kam's

Dangerous sound levels detected in last 10 minutes? yes

Flashing lights detected in last 10 minutes? yes

Ambient sound level: loud

Ambient light level: dark

Temperature: 78 F

BreadCo

Dangerous sound levels detected in last 10 minutes? no

Flashing lights detected in last 10 minutes? no

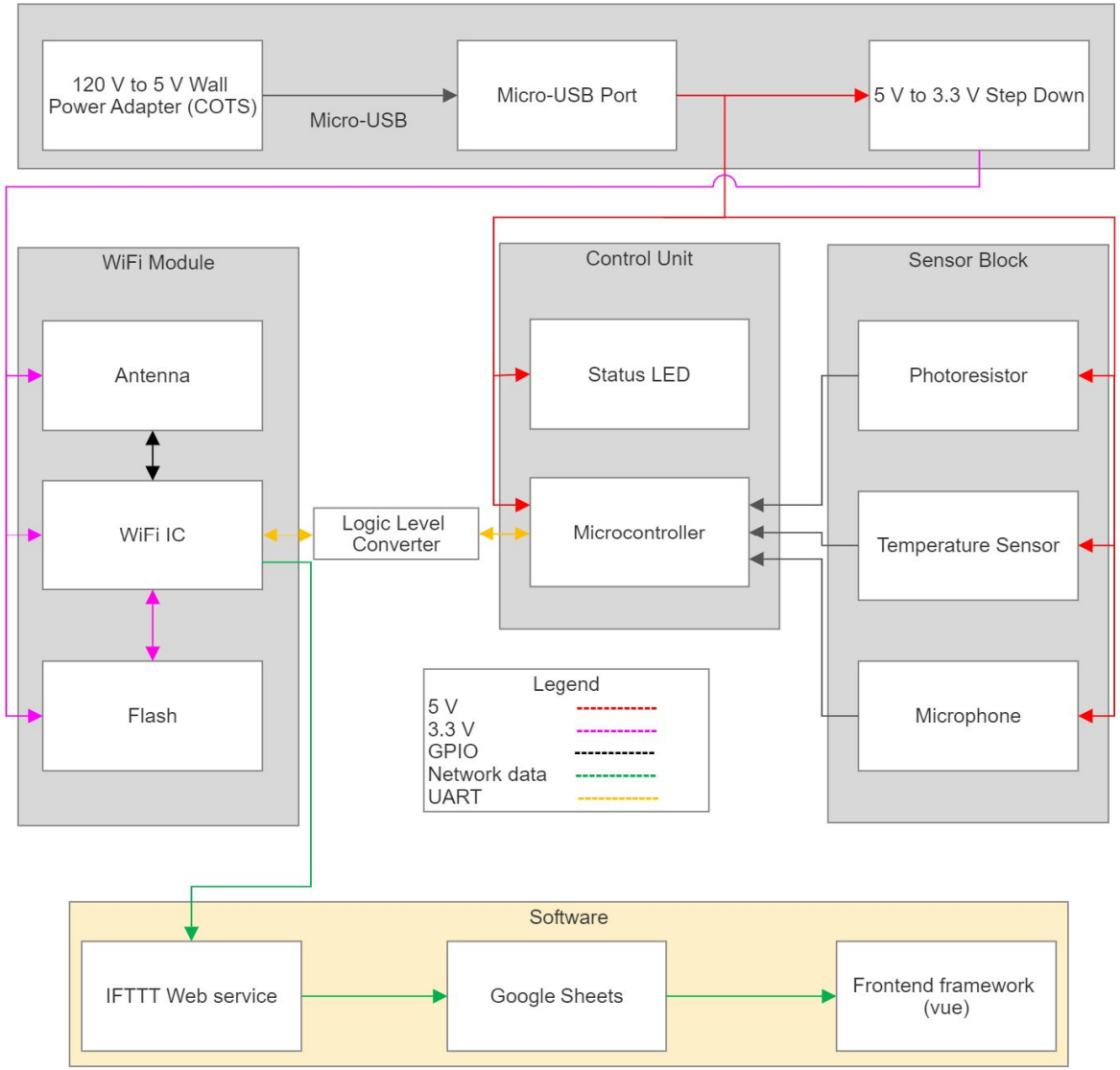
Ambient sound level: quiet

Ambient light level: dark

Final Product - Demonstration

Design and Build

Block Diagram

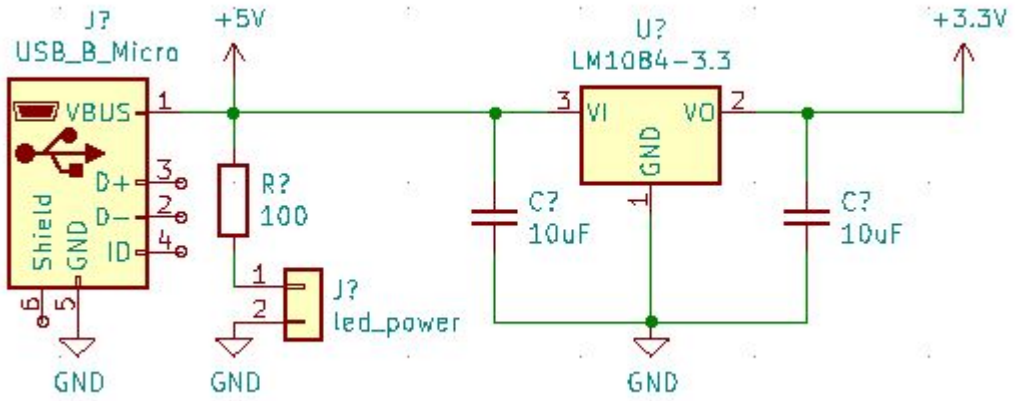


Power Supply

- Provides 0.5 A at 5 V +/- 0.5% from a wall power adaptor
- Provides 0.1 A at 3.3 V +/- 0.5% from a fixed LDO

Why LM1086-3.3?

- Needed an LDO since converting from only 5 V to 3.3 V
- Fixed 3.3 V output since we don't need to adjust our output

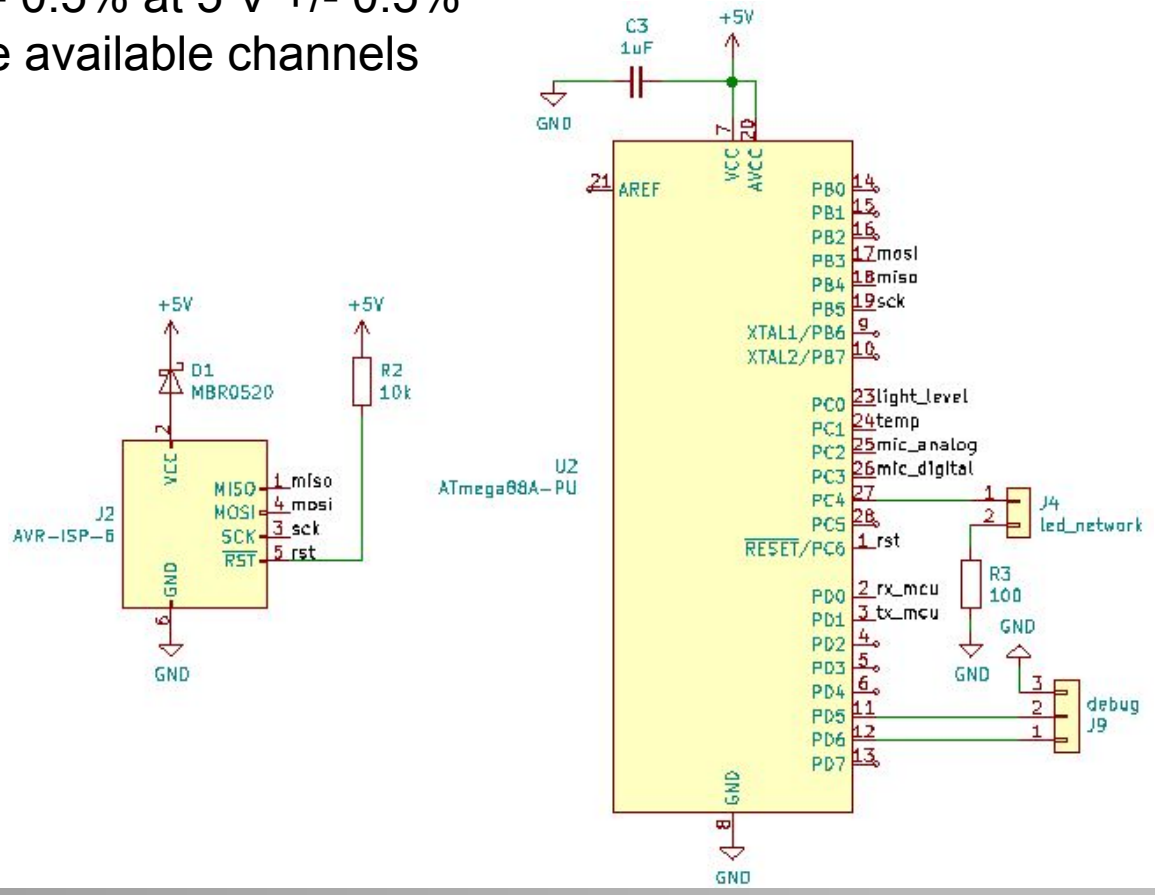


Control Unit

- Minimum of six GPIO pins that can sink/source 20 mA +/- 0.5% at 5 V +/- 0.5%
- A/D converter with at least ten bits of resolution and three available channels
- Must be able to communicate with 3.3 V UART signals

Why ATmega88A?

- Exceeded minimum specs (RAM, program memory)
- Through-hole packaging available

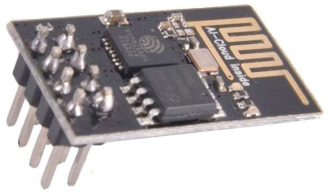


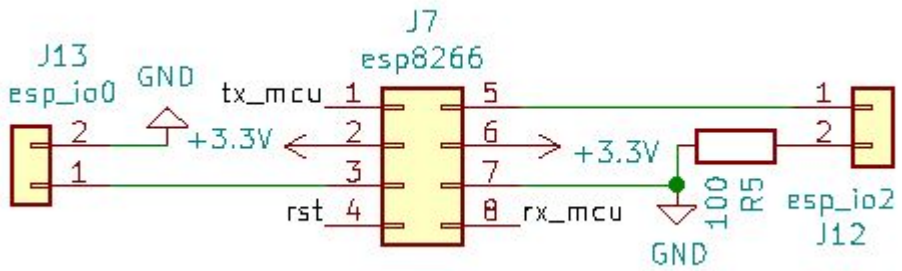
WiFi Module

- Must be able to send and receive data over UART
- Must be able to connect to WiFi network and send an HTTP request

Why ESP8266-01?

- Cheapest integrated package
- Exceeded minimum specs (RAM, program memory)
- Small footprint



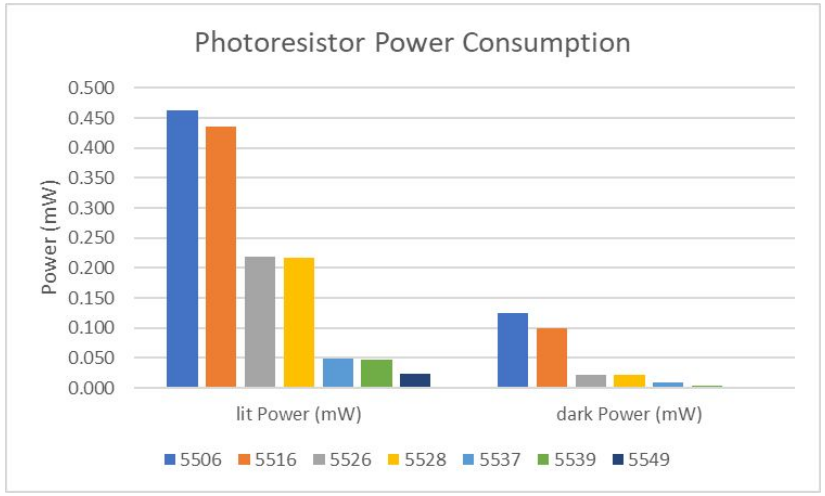


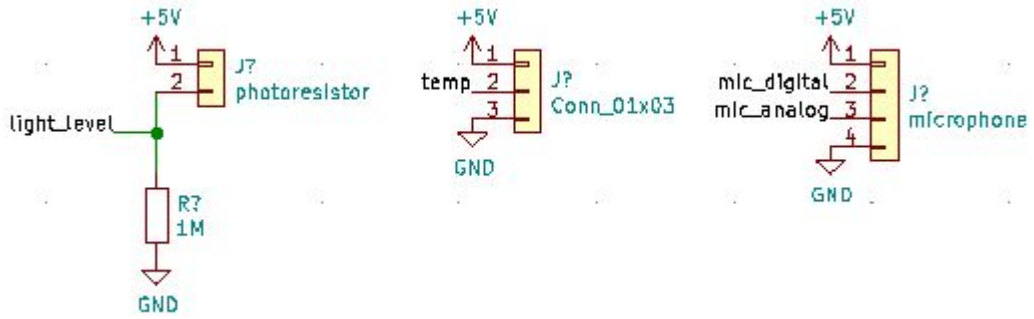
Sensor Block

- Sensor outputs must always be less than 20 mA and between 5 V and 0 V

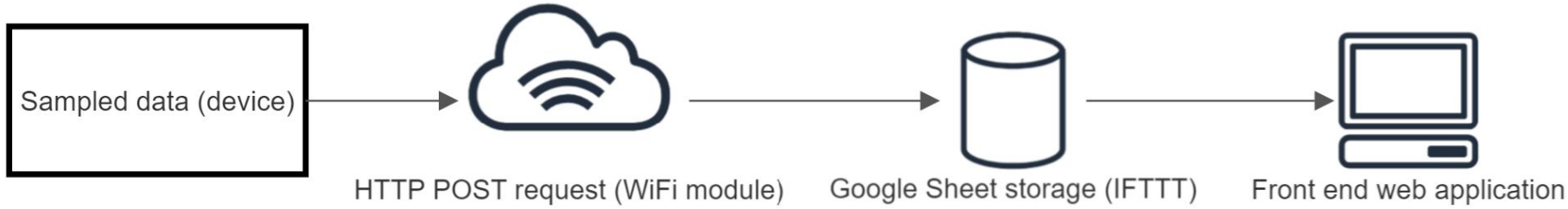
Why this photoresistor?

- In general, allowed for an analog reading (in contrast to photodiode or phototransistor)
- Photoresistor and accompanying resistor sized to maximize ADC usage while minimizing power consumption





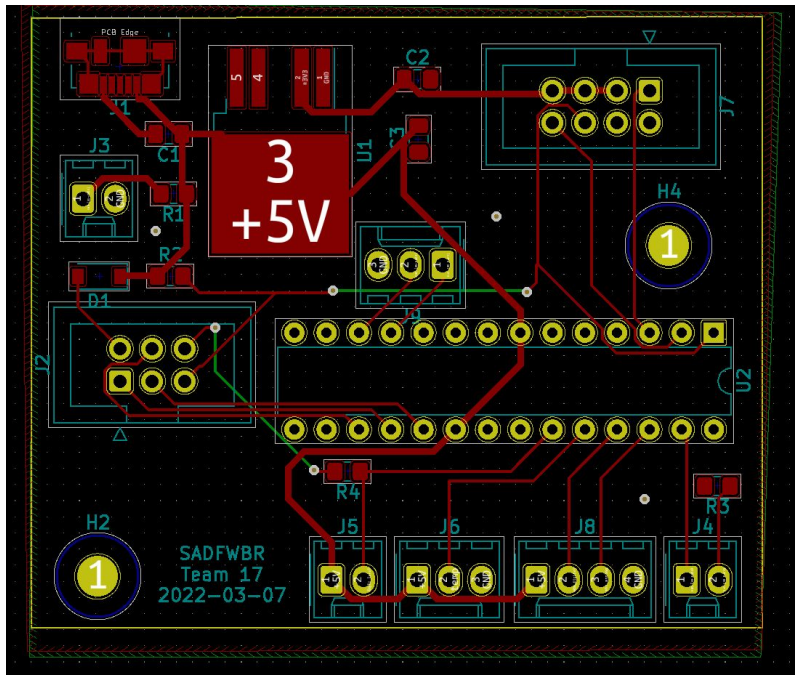
Software


Sampled data (device)
HTTP POST request (WiFi module)
Google Sheet storage (IFTTT)
Front end web application

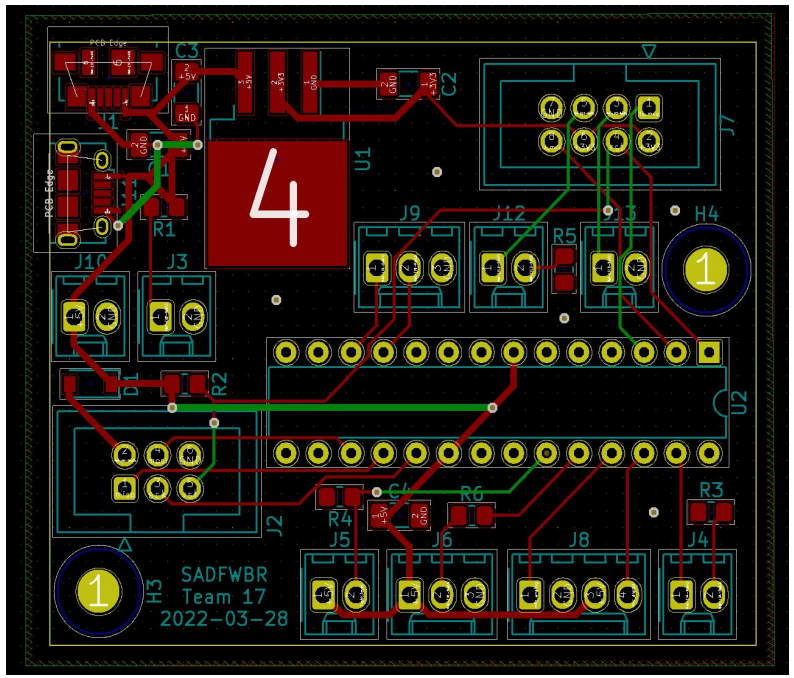
PCB Fabrication

Major changes from PCB 1 to PCB 2:

- Added additional connectors to the microcontroller and wifi chip for debugging
- Added a connector that could function as a power supply
- Changed mini-USB to micro-USB plug
- Corrected incorrect footprints (capacitor, LDO)

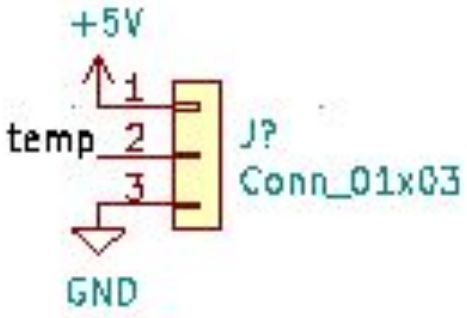


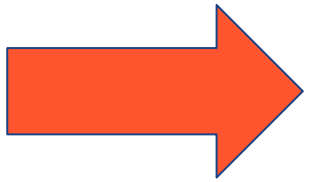
PCB 1

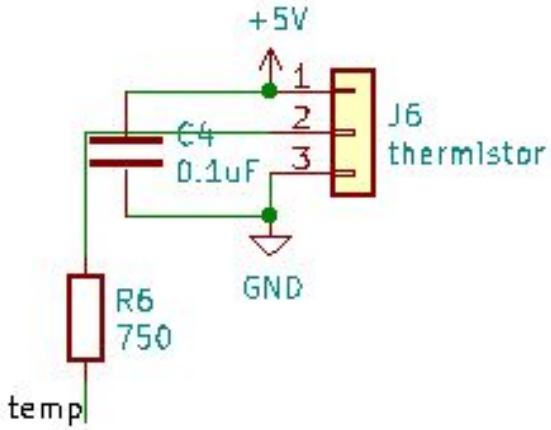


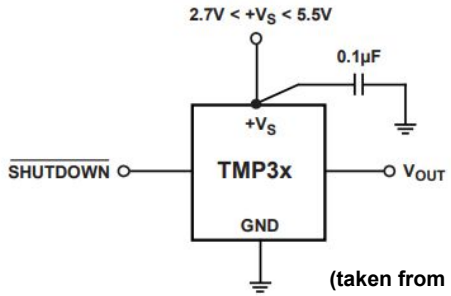
PCB 2

Temp Sensor Calibration







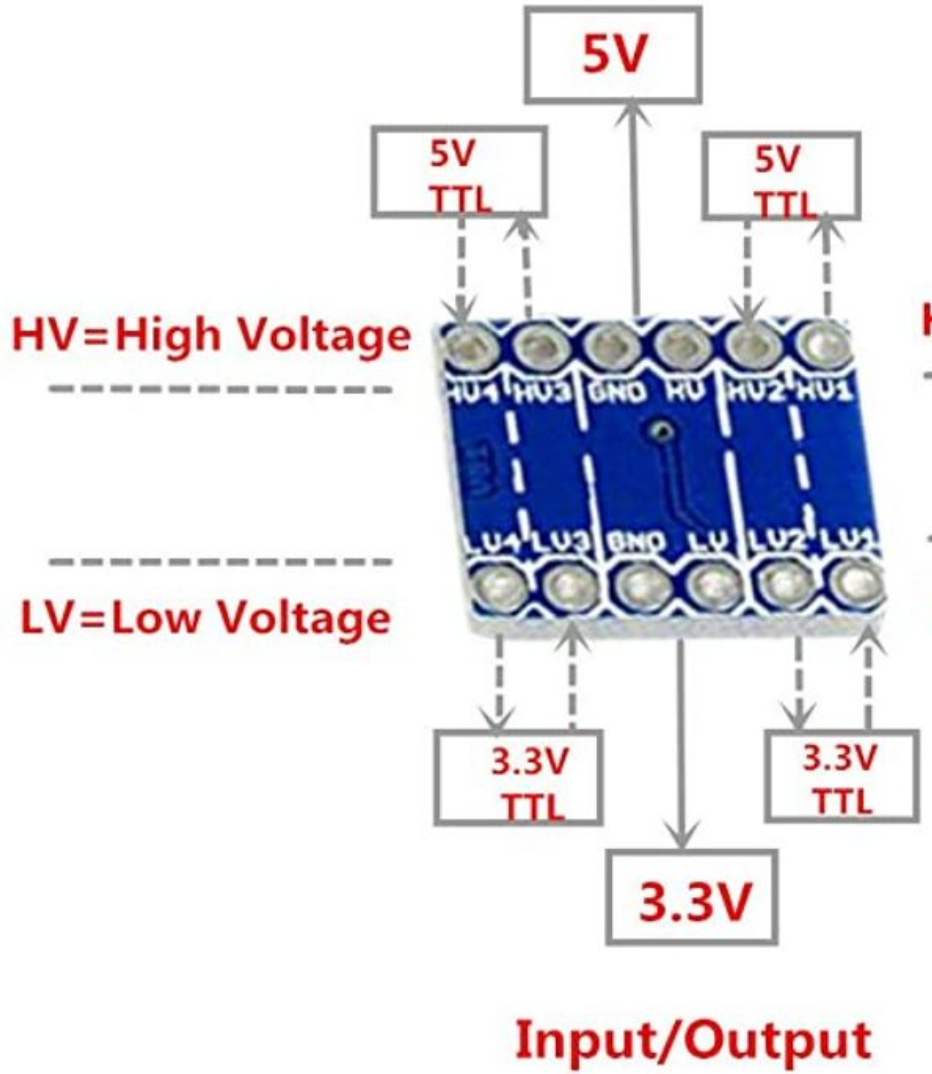


(taken from TMP36 datasheet)

Driving Factor:

Stability in temperature readings

Introduction of the Logic Level Converter



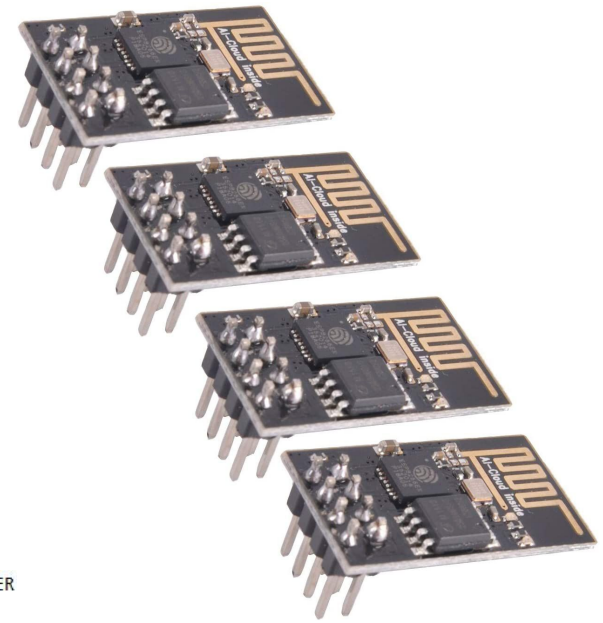
Logic Level Converter

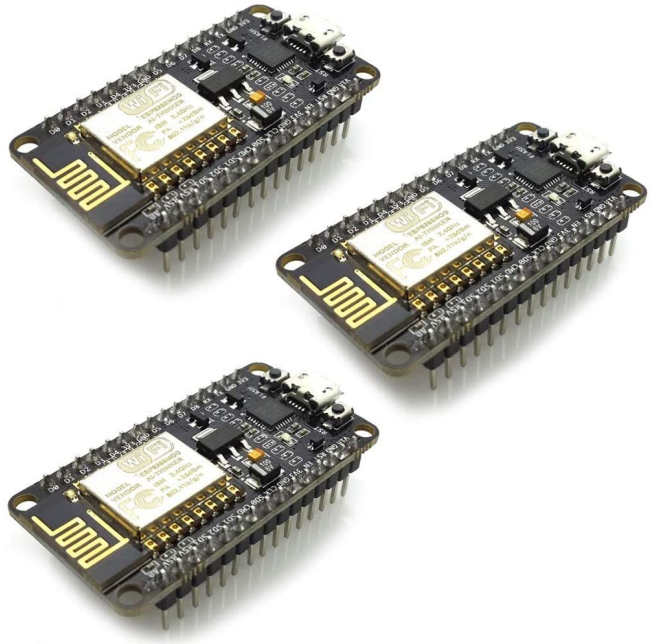
= Microcontroller Voltage

= WiFi Chip Voltage

WiFi Chip Substitution

ESP-01 WiFi Transceiver/Receiver





ESP-12E NodeMCU*

```
---------- CUT HERE FOR EXCEPTION DECODER

Exception (29):
epc1=0x4024e0e4 epc2=0x00000000 epc3=0x00000000 excvaddr=0x16e90000 depc=0x00000000

>>>stack>>>

ctx: sys
sp: 3fffed60 end: 3fffffb0 offset: 0190
3fffeef0:  4024e06d 00000000 3ffea6f4 3ffea700
3fffef00:  3ffea700 00000025 00000000 00000030
3fffef10:  00000000 3ffea70a 402618ef 3ffed220
3fffef20:  3ffea6f4 3fffdcc0 3ffe9848 000062e0
3fffef30:  000000d0 3ffed220 3fffdab0 00000000
3fffef40:  402611b3 3fffdab0 00000000 00000001
3fffef50:  3ffe9848 40000f49 3fffdab0 40000f49
3fffef60:  40000e19 00079599 00000000 00000005
3fffef70:  00000000 aa55aa55 000000ed 4010574d
3fffef80:  40105753 00000000 00000005 40100d1c
3fffef90:  4010000d 00796172 00079599 401000ac
3fffefa0:  402560ac 3fffef3c 40256065 3ffffd98
3fffefb0:  3fffffc0 00000000 00000000 feefeffe
3fffefc0:  feefeffe feefeffe feefeffe feefeffe
3fffefd0:  feefeffe feefeffe feefeffe feefeffe
```

Driving Factors:

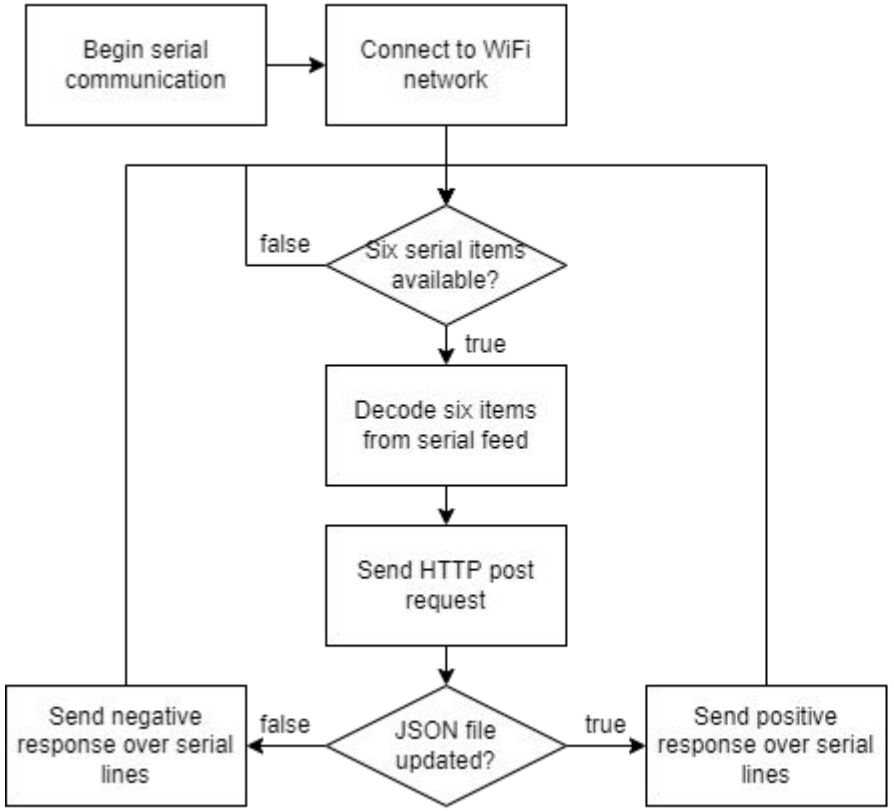
Avoid bricks, stable data transmission

* we ran the same code on these two devices with identical connections (i.e. we did not use any of the additional Development Board features)

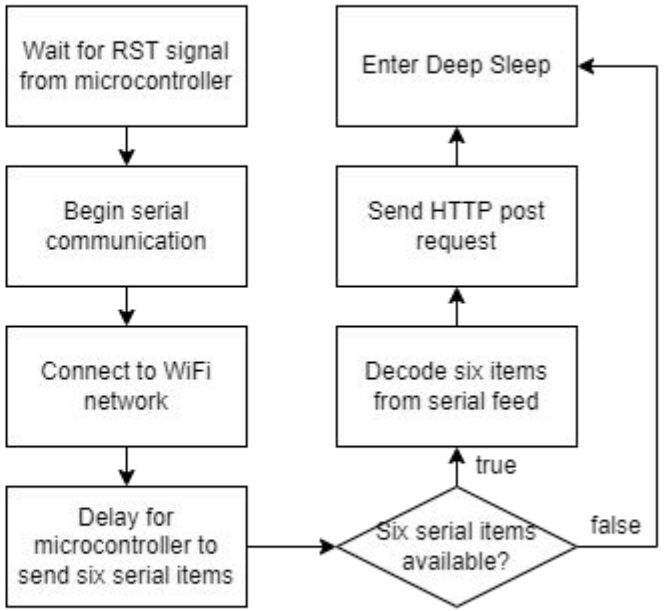
Embedded WiFi Chip Software

- Switched from the WiFi chip continuously polling to an “interrupt-like” method

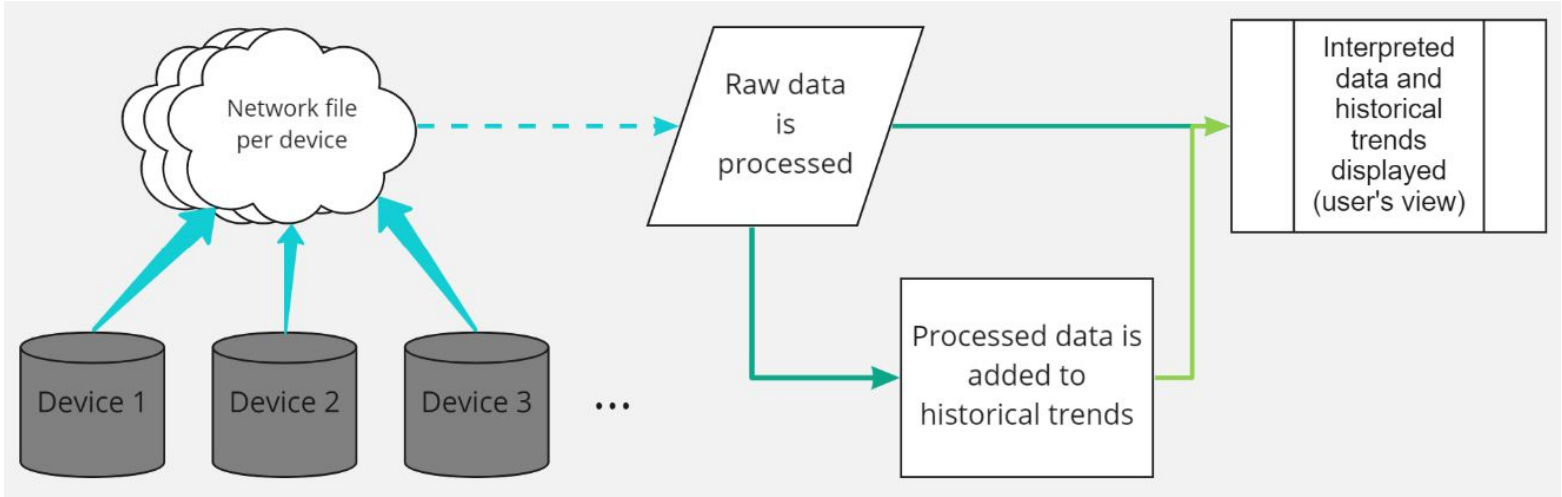
Continuous Polling

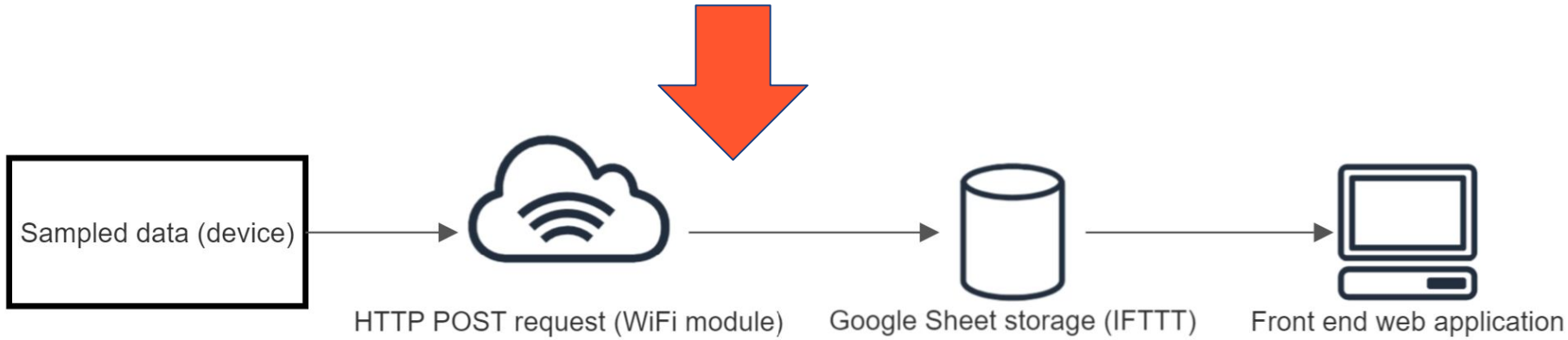


“Interrupt-like” Method



Higher Level Software





Conclusions

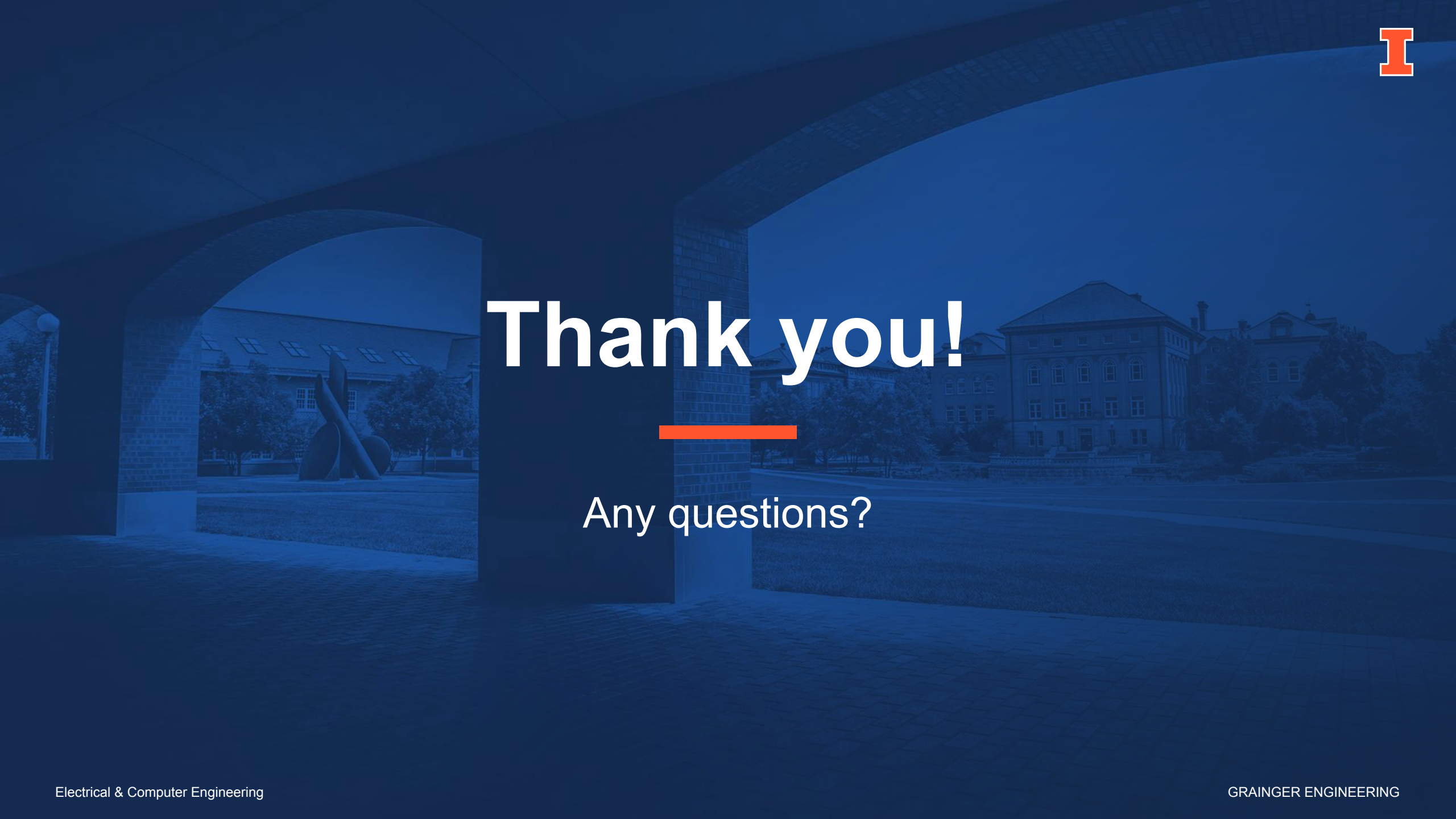
Requirements	Results	Requirements	Results
1. The power supply provides 5 V +/- 0.5% from a wall power adaptor. 2. The power supply provides a 3.3 V +/- 0.5% from a low dropout regulator driven by the 5V mentioned above. 3. The power supply is able to operate within 0-1 A from the 5 V source and able to operate within 0-0.1 A from the 3.3 V source.	passed	1. The C++ program must take raw input data from 3 sensors over the span of 10 minutes and output averaged levels in terms of temperature, light, and sound. 2. Data from the microcontroller must be able to be transmitted via the WiFi module to the JSON file. 3. The web application must display updated data within 1 minute +/- 30 seconds.	passed
1. There must be a total of six GPIO pins that can appropriately handle signals between 0 V and 5 V +/- 0.5% while sinking/sourcing at least 20 mA +/- 0.5% of current per pin. 2. The RX pin must be able to interpret 3.3 V +/- 0.5% signals (from WiFi chip) as logical HIGHs. 3. There must be an A/D converter with at least ten bits of resolution and three available channels which can be read sequentially.	passed	1. The output of the photoresistor must be within 0 V and 5 V +/- 0.5% with a current of no more than 20 mA +/- 0.5% in various light levels. 2. The microphone output must be within 0 V and 5 V +/- 0.5% with a current of no more than 20 mA +/- 0.5% depending on sound levels. 3. The temperature sensor output must be within 0 V and 5 V +/- 0.5% with a current of no more than 20 mA +/- 0.5% depending on temperature.	passed
1. The chip must be able to take commands via UART from an external microcontroller. 2. The RX pin must be able to handle a 5 V +/- 0.5% signal (from the microcontroller) without breaking the chip. 3. The chip must be able to connect to a network and send 4 kB (max size) of data supplied by the microcontroller once every ten minutes +/- thirty seconds.	passed		

What We Accomplished

- Safety alerts identified
- Ambient sound and light levels classified
- Enclosure completed (25% of goal size)
- Data interpretation and display on frontend (down from 10 minutes to 1 minute)

Going Further

- Use more powerful components
- Add music genre identification
- Reduce words on webpage
- Integrate location information into web application
- Allow for interaction and feedback to web application


Thank you!
Any questions?

The Grainger College of Engineering

UNIVERSITY OF ILLINOIS URBANA-CHAMPAIGN

Appendices

ATMega88A Code

```
void setup() {

  Serial.begin(115200);

  pinMode(micD, INPUT);
  pinMode(micA, INPUT);
  pinMode(tempSensor, INPUT);
  pinMode(lightSensor, INPUT);
  pinMode(rstPin, OUTPUT);
  digitalWrite(rstPin, HIGH);
}

void loop() {

  int buf = 0;
  int util = 0;
  int maxVol = 0;

  digitalWrite(rstPin, LOW);
  delay(20);
  digitalWrite(rstPin, HIGH);
  delay(180);

  // Temperature Signal
  for(int i = 0; i < 16; i++) {
    buf = buf + analogRead(tempSensor);
  }

  buf = buf >> 4; // average

  sendChar((char)(buf)); // chop off two LSb's to allow a send over one byte
  sendChar((char)(buf >> 8)); // chop off two LSb's to allow a send over one byte
```

```
// Microphone Signals
for(int i = 0; i < 3000; i++) {
  buf = analogRead(micA);
  if(buf > maxVol) maxVol = buf;
}

sendChar((char)(maxVol >> 2));

buf = 0;
for(int i = 0; i < 1024; i++) {
  if(digitalRead(micD))
    buf = buf + 1;
}

if(buf > 4)
  sendChar('1');
else
  sendChar('0');

// Photoresistor Signals
buf = 0;
maxVol = analogRead(lightSensor);
for(int i = 0; i < 16; i++) {
  int x = analogRead(lightSensor);
  buf = buf + x;
  if(abs(maxVol - x) > 50)
    util++;
  delay(30);
}
buf = buf >> 4;

sendChar((char)(buf >> 2));

if(util > 1)
  sendChar('1');
else
  sendChar('0');

sendChar('\n');

delay(10 * 1000);
```

ESP8266 WiFi Code

```
#include <ESP8266WiFi.h>

// WiFi Connection Info
const char* ssid     = "IllinoisNet_Guest";

// IFTTT Webhooks
const char* server = "maker.ifttt.com";
const char* resource = "/trigger/partybot/with/key/pN

int temp = 0;
int noise = 0;
int light = 0;

void setup() {

  Serial.begin(115200);
  delay(20);

  initWifi();

  serialDecode();

  httpRequest();

  ESP.deepSleep(0);

}

void loop() {
  // will never get here because we deep sleep first
}
```

```
void initWifi() {

  WiFi.begin(ssid);

  while(WiFi.status() != WL_CONNECTED) {
    yield();
  }

}
void httpRequest() {

  WiFiClient client;
  int retries = 5;
  while(!!!client.connect(server, 80) && (retries-- > 0)) {
    delay(2);
  }


  float trueTemp = (((float)temp * 5.0 / 1024.0) - 0.5) * 100.0 * 9.0 / 5.0 + 32.0;

  String jsonObject = String("{\"value1\":\"") + trueTemp + "\",\"value2\":\"" + noise
                      + "\",\"value3\":\"" + light + "\"}";

  client.println(String("POST ") + resource + " HTTP/1.1");
  client.println(String("Host: ") + server);
  client.println("Connection: close\r\nContent-Type: application/json");
  client.print("Content-Length: ");
  client.println(jsonObject.length());
  client.println();
  client.println(jsonObject);

  int timeout = 5 * 10; // 5 seconds
  while(!!!client.available() && (timeout-- > 0)){
    delay(100);
  }

  client.stop();

}
```

```
void serialDecode() {

  char buf[7];

  while(Serial.available() < 7) {
    delay(2);
  }

  buf[0] = Serial.read();
  buf[1] = Serial.read();
  buf[2] = Serial.read();
  buf[3] = Serial.read();
  buf[4] = Serial.read();
  buf[5] = Serial.read();
  buf[6] = Serial.read();

  temp = (((int)buf[1]) << 8) + ((int)buf[0]);

  noise = ((int)buf[2]) << 2;
  if(buf[3] != '0')
    noise += 1;

  light = ((int)buf[4]) << 2;
  if(buf[5] != '0')
    light += 1;

  while(Serial.available()) {
    Serial.read();
  }
}
```

What We Learned

- PCB best practices
- How to choose components online
- How to integrate sensor data into web design

HEADING

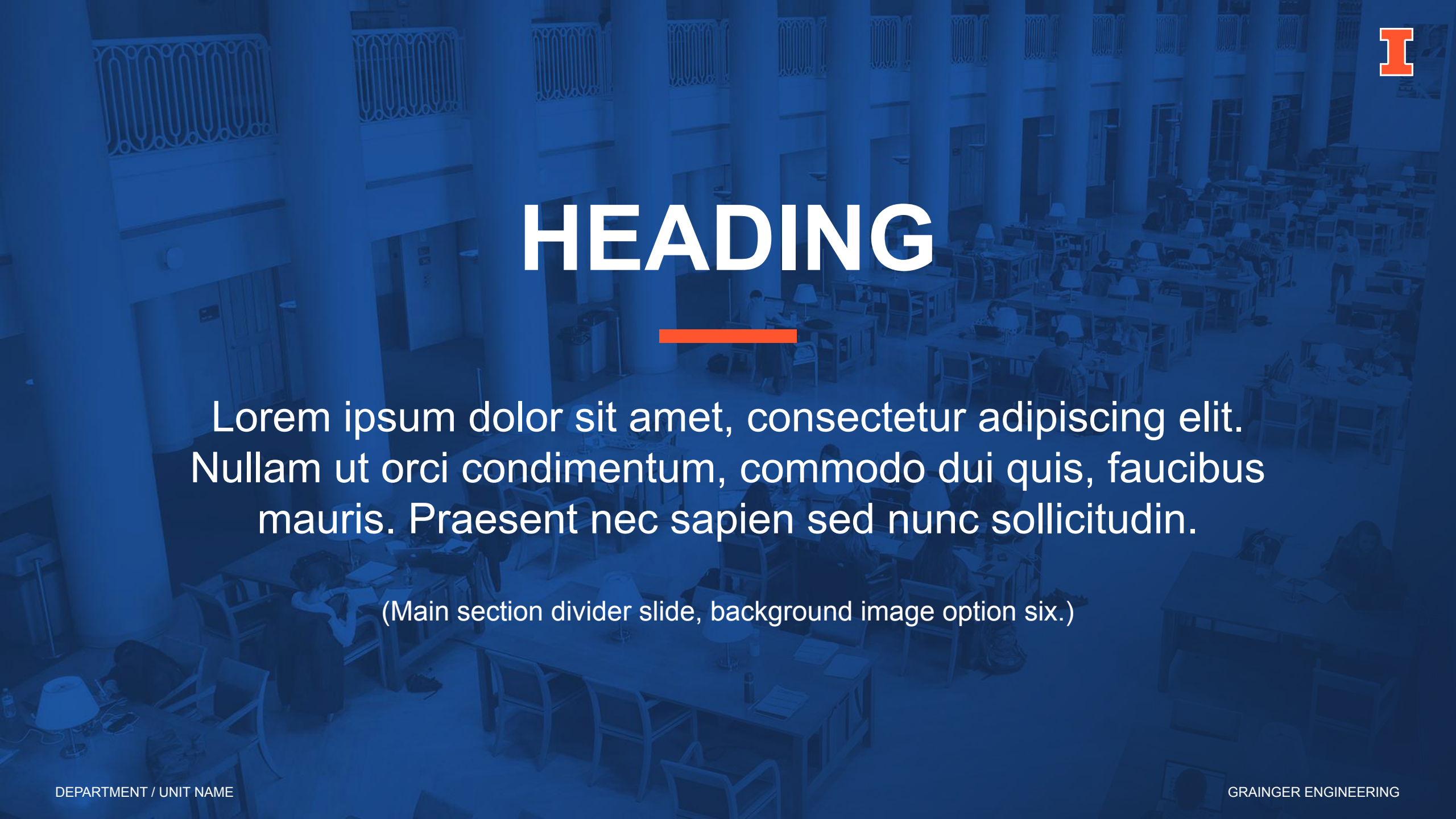
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ut orci condimentum, commodo dui quis, faucibus mauris. Praesent nec sapien sed nunc sollicitudin.

(Main section divider slide, background image option five.)

HEADING

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ut orci condimentum, commodo dui quis, faucibus mauris. Praesent nec sapien sed nunc sollicitudin.

(Main section divider slide, background image option one.)



HEADING

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam ut orci condimentum, commodo dui quis, faucibus mauris. Praesent nec sapien sed nunc sollicitudin.

(Main section divider slide, background image option six.)

Heading

(Transition slide option two.)

Lorem Ipsum Dolor Sit Amet Consectetur Adipiscing

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem Ipsum Dolor Sit Amet Consectetur Adipiscing

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

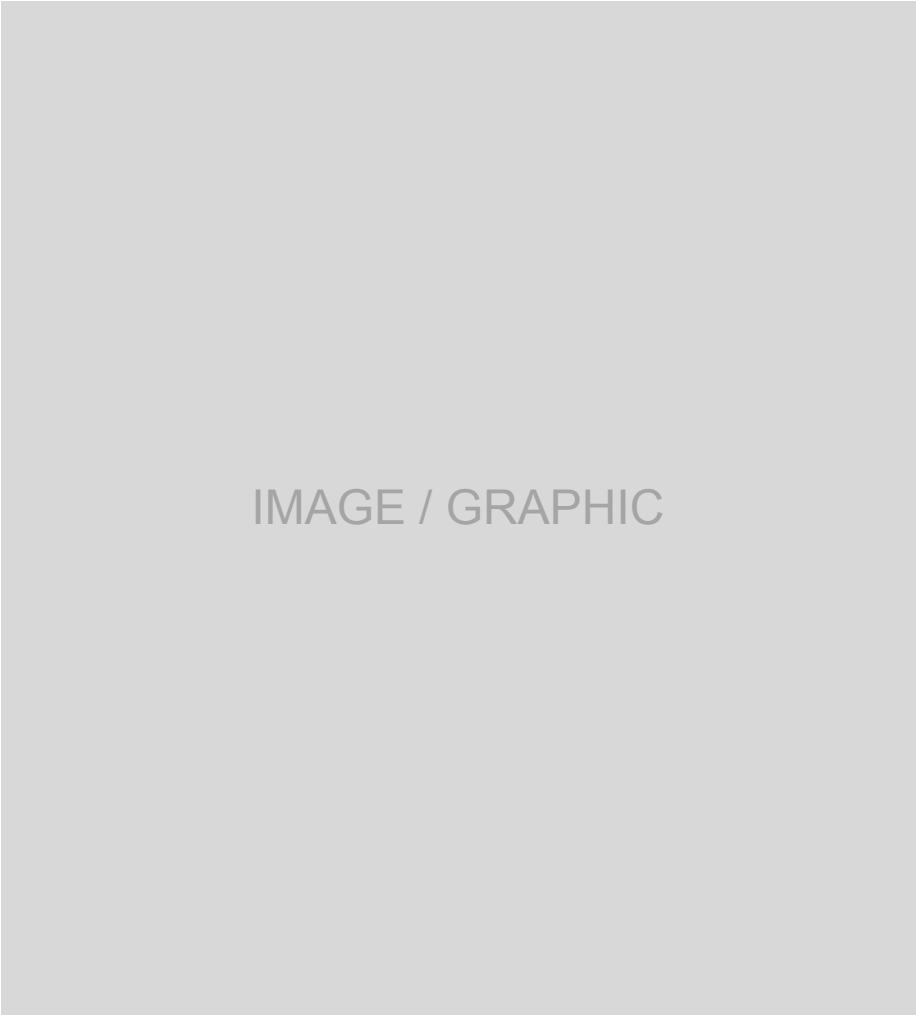
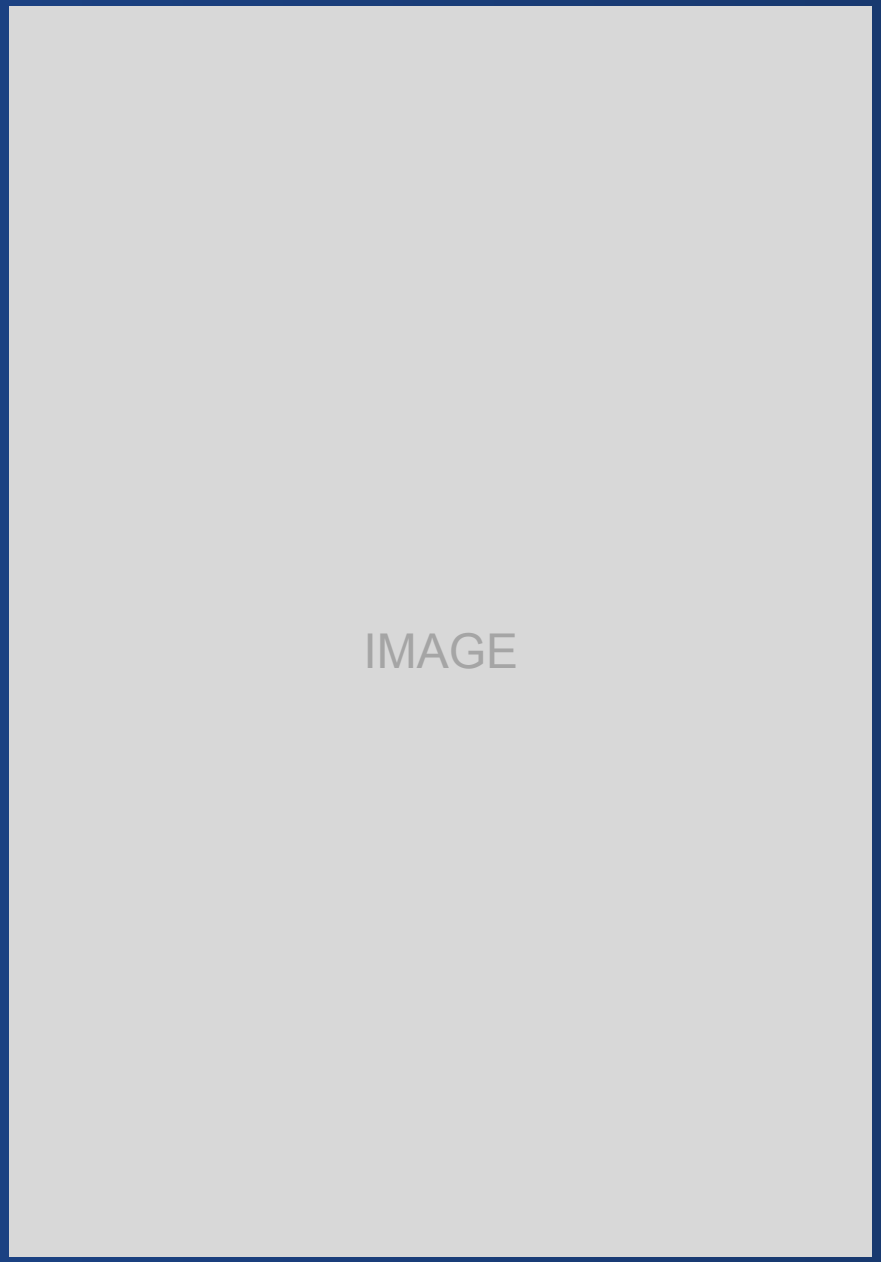


IMAGE (Send to back so Block I is not covered.)

(Text here relevant to the photo for a larger call out on a given idea or message.)
(Keep this text simple and short on one or two lines.)

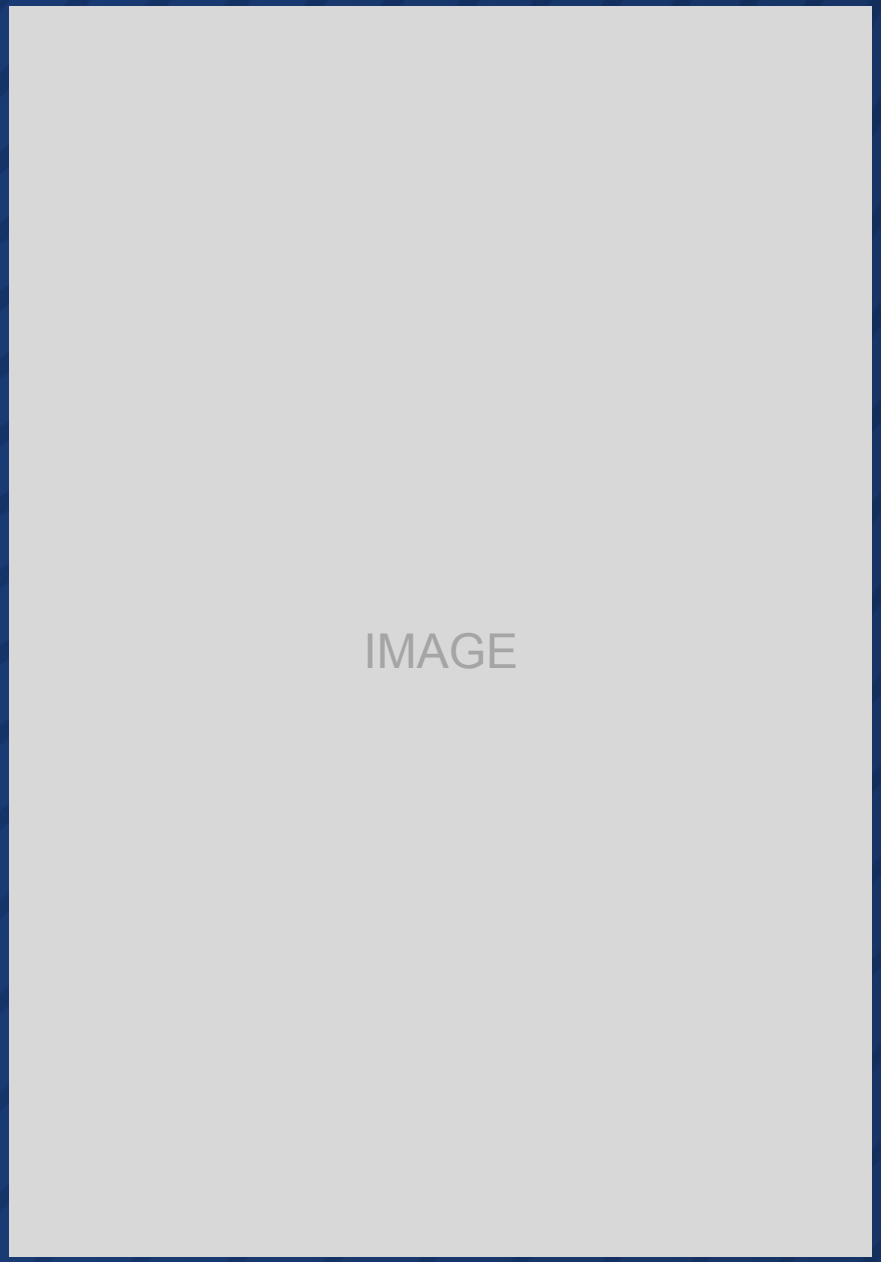


Lorem ipsum dolor sit amet consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet consectetur adipiscing

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.



Lorem ipsum dolor sit amet consectetur adipiscing elit

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Lorem ipsum dolor sit amet consectetur adipiscing

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

IMAGE

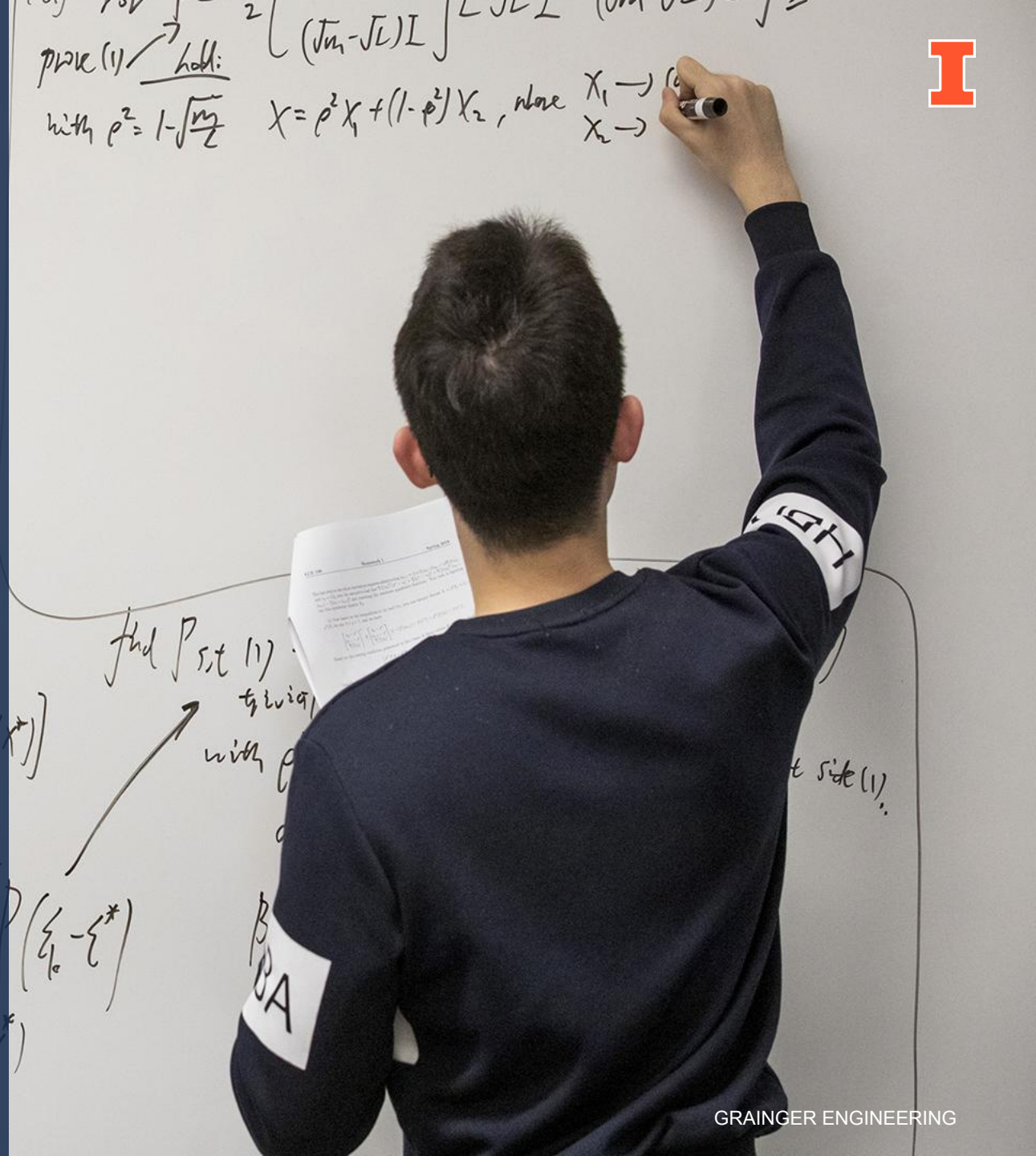
(Use a photo for the entire slide like the example shown here. Send the photo to the back behind the blue shaded box. Be sure the Block I is not covered up and is in front of both the photo and blue shaded box.)

(Text here relevant to the photo for a larger call out on a given idea or message. Keep this text simple and short on one or two lines.)

IMAGE

(Use a photo for the entire slide like the example shown here. Send the photo to the back so the Block I and footer text is not covered.)

(Slide the blue transparent box and the text left or right to fit over your background image. Use this text box for a call out or caption to the image.)



(Closing slide option one. Suggestions for last slide)

Summary
Thank You
Questions
Contact Information

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

(Closing slide option two. Suggestions for last slide)

Summary
Thank You
Questions
Contact Information

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.