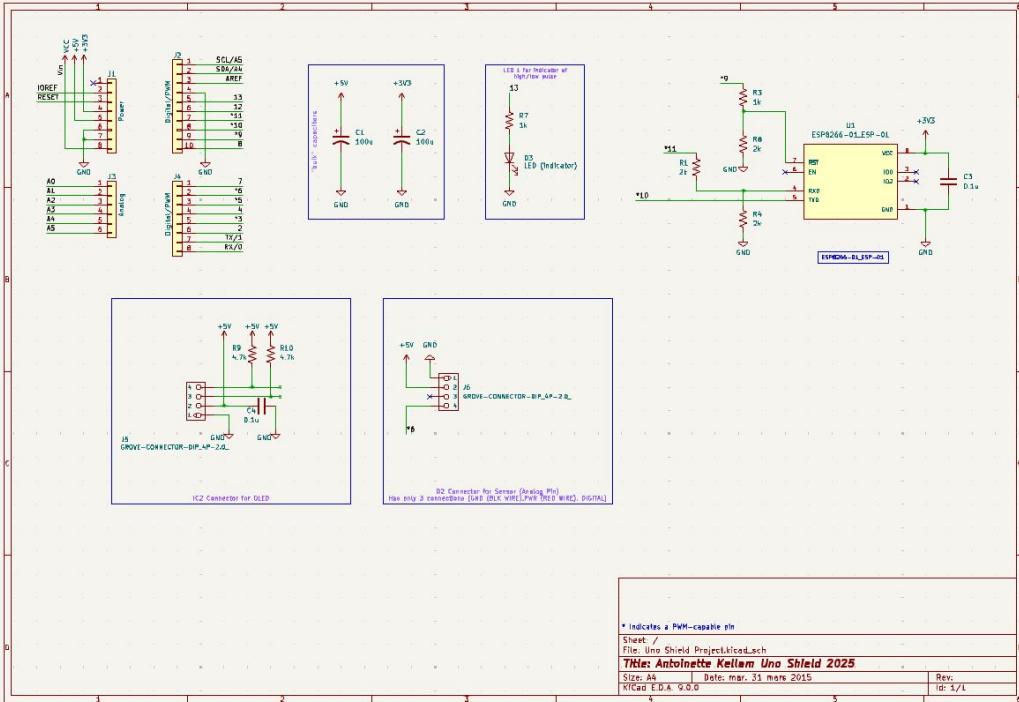




PulseSync IoT: Real-Time Heart-Rate Monitor w/LED Synchronization

Designed/Tested by Antoinette Kellam - CPEG298

Schematic & Circuit Design



The KiCad schematic (.kicad_sch) of my IoT Heart Rate device is shown. The Arduino UNO connections, like power and I/O headers, are shown.

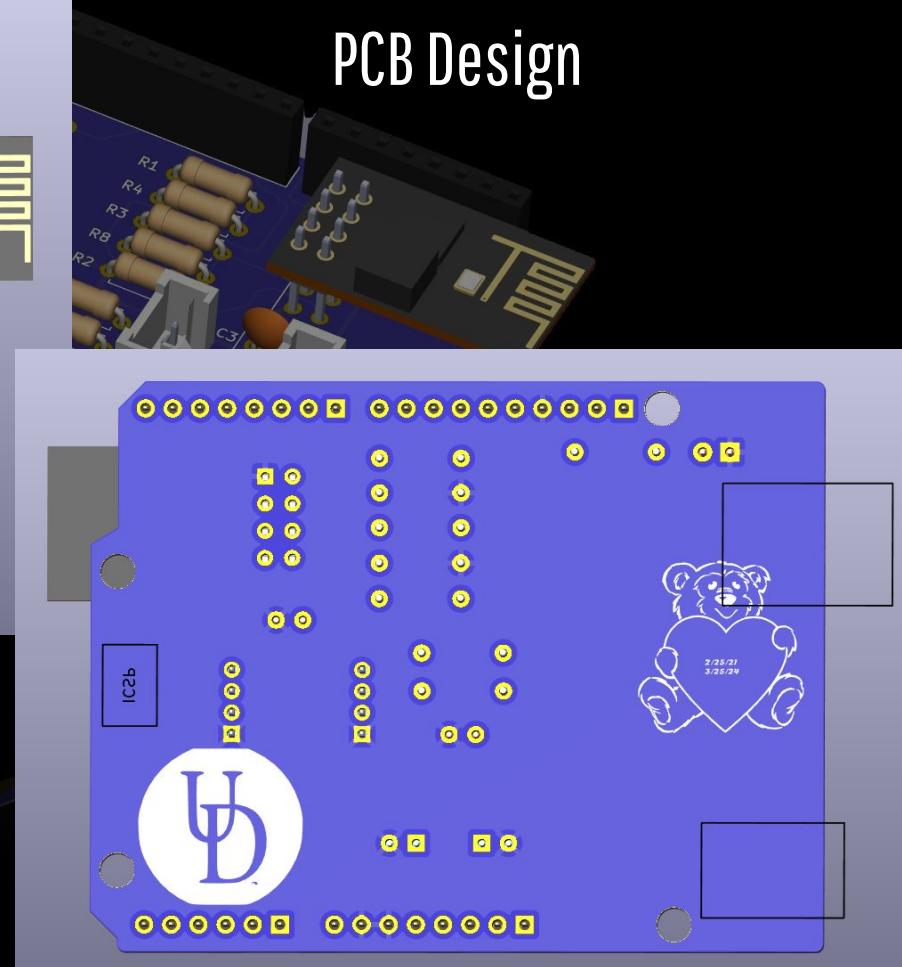
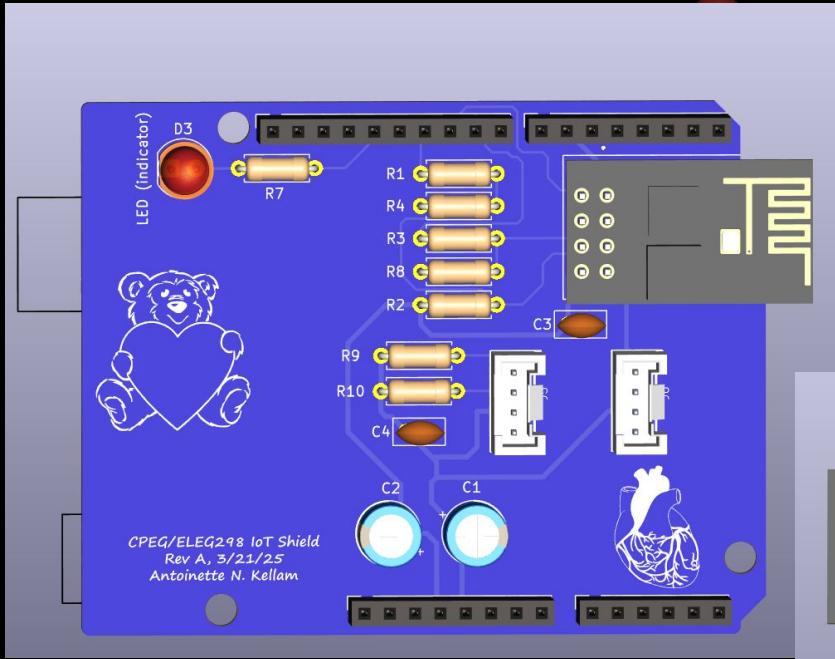
In the top-center, I have 2 bulk capacitors (100uF on 5V and +3V3 rails).

The upper-far-right displays the ESP8266-01 WiFi module with its 3.3V bypass capacitors and resistors that are in connections to the Arduino Uno 5V UART and EN lines.

The lower-left houses the Grove I2C port connector for an OLED screen that could be used to display the current bpm (*this was not functioning properly).

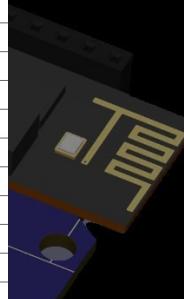
Centered below my “bulk capacitors” is the Grove Senor port that houses the main sensor used in this project. This was a 3-wire connector for the Grove-Ear-Clip Heart Rate Sensor: GND, PWR, and Digital Pin that supplies 5V and GND.

PCB Design



BOM DIAGRAM

Id	Designator	Footprint	Quantity	Designation	Supplier and Ref
1	J1	PinSocket_1x08_P2.54mm_Vertical	1	Power	
2	J3	PinSocket_1x06_P2.54mm_Vertical	1	Analog	
3	J2	PinSocket_1x10_P2.54mm_Vertical	1	Digital/PWM	
4	J4	PinSocket_1x08_P2.54mm_Vertical	1	Digital/PWM	
5	R4,R1,R2,R8	R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal	4	2k	
6	U1	XCVR_ESP8266-01_ESP-01	1	ESP8266-01_ESP-01	
7	C3, C4	C_Disc_D5.0mm_W2.5mm_P2.50mm	2	0.1u	
8	D3	LED_D5.0mm	1	LED (indicator)	
9	J6, J5	GROVE-HW4-2.0	2	GROVE-CONNECTOR-DIP_4P-2.0	
10	R7, R3	R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal	2	1k	
11	R10,R9	R_Axial_DIN0207_L6.3mm_D2.5mm_P7.62mm_Horizontal	2	4.7k	
12	C2, C1	CP_Radial_D6.3mm_P2.50mm	2	100u	



The parts were sourced from the University of Delaware undergraduate labs and DigiKey.
The unique parts and prices are shown in the table below:

Component	DigiKey part #	Price/each	Description
ESP8266-01/ESP-01	3647-Ai-ThinkerESP-01S ESP8266-ND	\$3.96	Ai-Thinker ESP-01S ESP8266 Wi-Fi
ADS1115IDGS	296-38849-1-ND	\$5.30	IC ADC 16BIT SIGMA-DELTA 10VSSOP
Conn_01x04	1597-1082-ND	\$1.70 [10]	GROVE 2MM 4PIN VERT CONN 10PCS
SJ1-3535NG	CP1-3535NG-ND	\$1.23	CONN JACK STEREO 3.5MM R/A
PJ-047AH	CP-047A-ND	\$0.88	CONN PWR JACK 2X5.5MM SOLDER
SK6812 NeoPixel	1528-1104-ND	\$4.06 [10]	ADDRESS LED DISC SERIAL RGB 1=10
Grove DHT20 sensor	1597-101020932-ND	\$5.86	GROVE TEMP+HUM SENSOR V2.0 DHT20
Grove 0.96" OLED	1597-104020208-ND	\$4.96	GROVE - OLED DISPLAY 0.96" (SSD1313)
YHDC SCT013-000	1597-1307-ND	\$10.36	CURR SENSE XFMR 100A:0.05A 100A
WAU12-200	237-1880-ND	\$9.03	AC/AC WALL MNT ADPT 12VAC 200MA

ARDUINO CODE HIGHLIGHTS

```
101 void setup() {
102     // - Serial setups -
103     Serial.begin(115200); // for the monitor
104     espSerial.begin(9600); // for the commands
105
106     // - Reset ESP-01S -
107     pinMode(RESET_PIN, OUTPUT);
108     digitalWrite(RESET_PIN, LOW);
109     delay(1000);
110     digitalWrite(RESET_PIN, HIGH);
111     delay(2000); // 2 sec loop reading
112
113     // - Sensor & LED pins -
114     pinMode(HEART_SENSOR_PIN, INPUT_PULLUP);
115     pinMode(LED_PIN, OUTPUT);
116
117     // - Drain any stray ESP bytes -
118     unsigned long t0 = millis();
119     while (millis() - t0 < 2000) {
120         while (espSerial.available()) espSerial.read();
121         pollHeartSensor();
122     }
123
124     // - Adafruit IO handshake -
125     Serial.println("\n\n--- Heart Rate + Adafruit IO ---");
126     espSend("get_macaddr");
127     espSend("get_version");
128     espSend("wifi_ssid=" + WIFI_SSID);
129     delay(500);
130     espSend("wifi_pass=" + WIFI_PASS);
131     delay(500);
132     espSend("io_user=" + IO_USERNAME);
133     espSend("io_key=" + IO_KEY, 2000, false);
134
135
136     String r = espSend("setup_io", 30000);
137     if (r.indexOf("connected") < 0) {
138         Serial.println("Adafruit IO Connection Failed");
139         while (1);
140     }
141     espSend("setup_pubfeed=1,HeartRate");
142     Serial.println("----- Setup Complete -----");
143
144 }
```

```
145 void loop() {
146     // always poll sensor first (handles LED off-timer too)
147     pollHeartSensor(); //edge detects rising pulse from D6
148
149     if (newReady) {
150         // send immediately on each beat
151         Serial.print("Sending BPM: "); Serial.println(heartRate);
152         espSerial.print("send_data=1,");
153         espSerial.println(heartRate);
154
155         // wait up to 2s for ESP reply, but keep polling sensor:
156         String resp;
157         unsigned long t1 = millis();
158         while (millis() - t1 < 2000) {
159             while (espSerial.available()) {
160                 resp += char(espSerial.read());
161             }
162             pollHeartSensor();
163         }
164         Serial.print("Resp: "); Serial.println(resp);
165
166         newReady = false;
167         // no extra delay - each heartbeat triggers its own send & flash
168     }
169 }
170 }
```

ESP_01S_AdafruitIO_demoONCEAGAIN | Arduino IDE 2.3.6

File Edit Sketch Tools Help



Arduino Uno

ESP_01S_AdafruitIO_demoONCEAGAIN.ino

```
150 espSerial.print("send_data=1,");
151 espSerial.println(heartRate);
152
153 // wait up to 2s for ESP reply, but keep polling sensor:
154 String resp;
155 unsigned long t1 = millis();
156 while (millis() - t1 < 2000) {
157   while (espSerial.available()) {
158     resp += char(espSerial.read());
159   }
160   pollHeartSensor();
161 }
162 Serial.print("Resp: "); Serial.println(resp);
163
164 newReady = false;
165 // no extra delay - each heartbeat triggers its own send & flash
166
167 }
168 }
```

Output Serial Monitor X

Message (Enter to send message to 'Arduino Uno' on 'COM3')

Resp: sent-> HeartRate,80.00000

Sending BPM: 82

Resp: sent-> HeartRate,82.00000

Sending BPM: 87

Resp: sent-> HeartRate,87.00000

Sending BPM: 82

Resp: sent-> HeartRate,82.00000

Sending BPM: 85

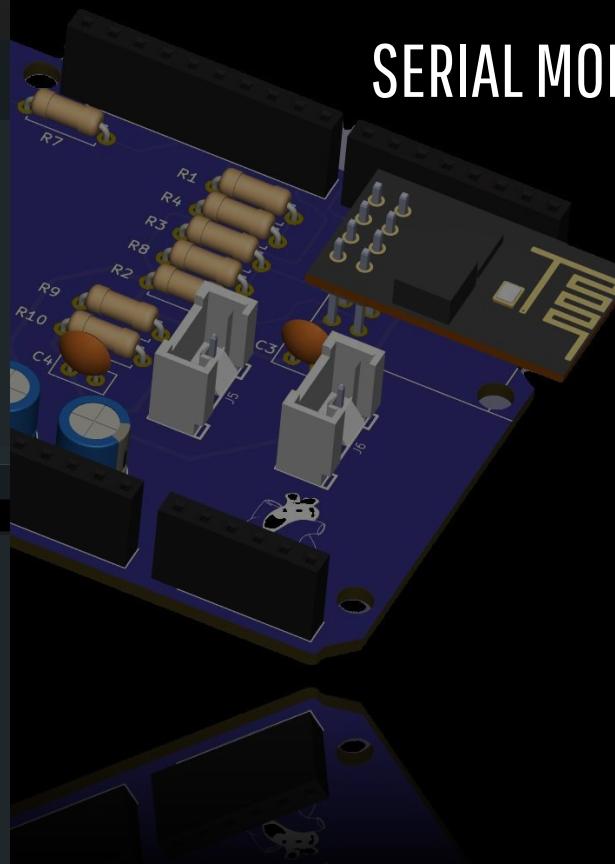
Resp: sent-> HeartRate,85.00000

Sending BPM: 87

Resp: sent-> HeartRate,87.00000

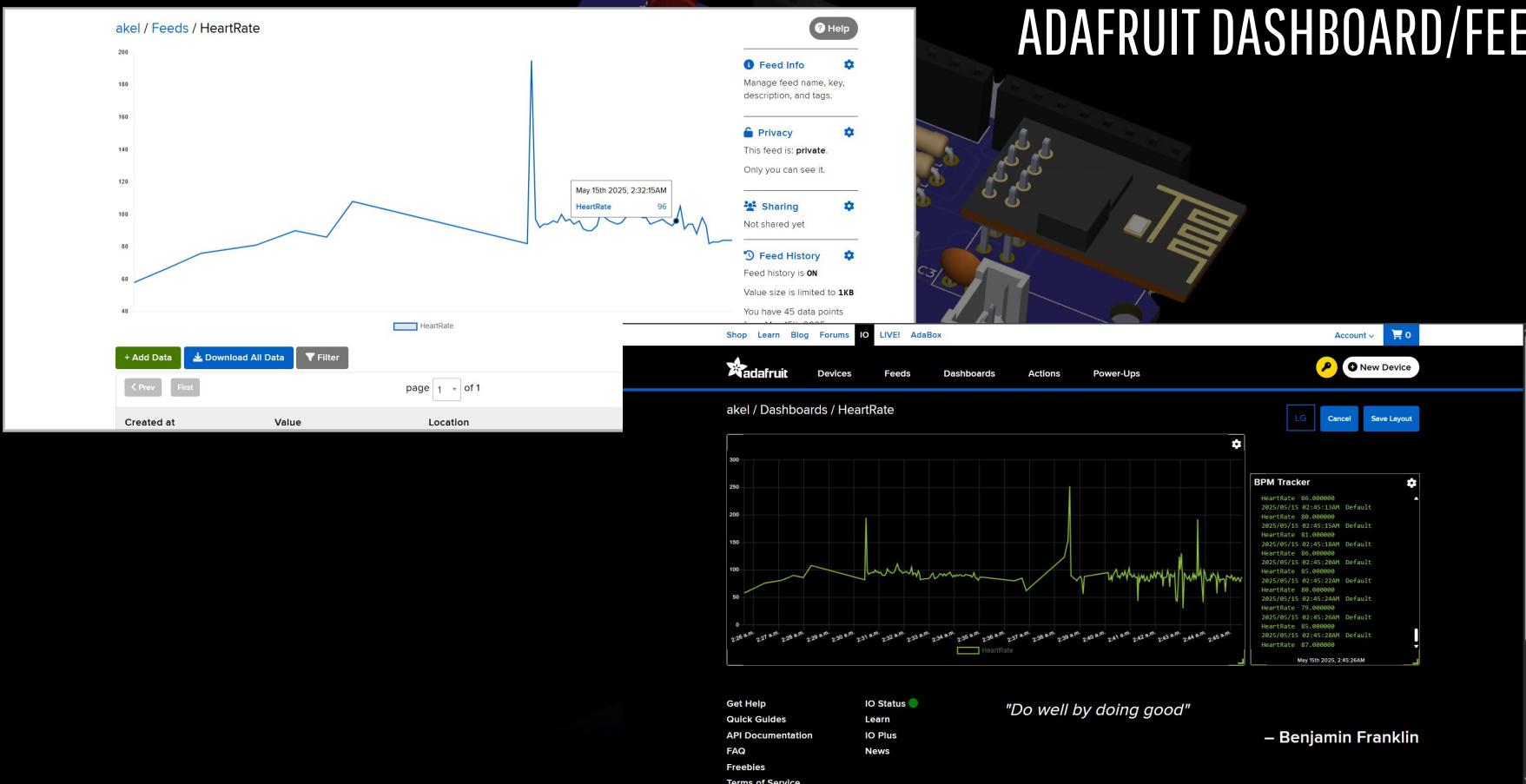


Sending BPM: 85



SERIAL MONITOR/PLOTTER

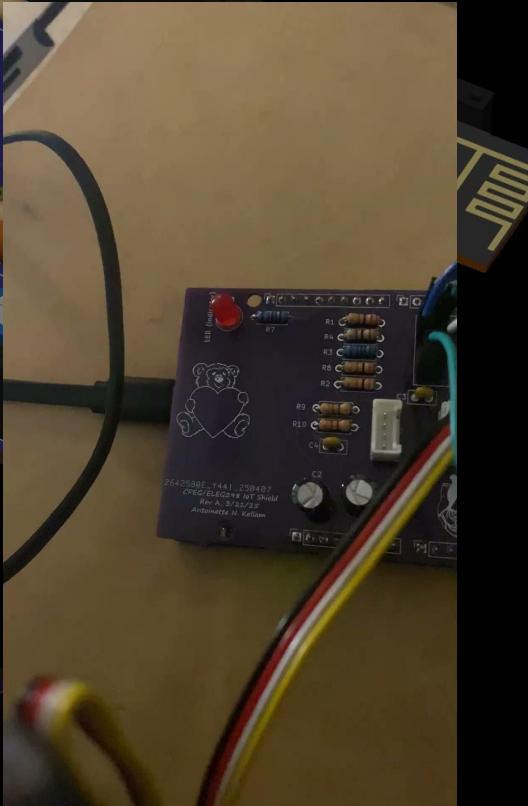
ADAFRUIT DASHBOARD/FEED



SUPPLEMENTARY VIDEOS



HEART RATE SENSOR (ALONE WORKING)



COMPLETE PROJECT WORKING (SENDING DATA VIA WIFI MODULE)

DEMO!

