Designing Within Reach

A Wearable Heart Rate Monitor Constrained by Available Fabrication

Engineering Design Report

Author Wilson Conway
School Briarcrest Christian School
Project Dates Build: July 21 – July 31, 2026 · Testing: August 1 – August 3, 2026

Table of Contents

A Note on Structure and Voice

Sections 3 through 7, 10, and 12 follow formal third-person engineering-report convention. Sections 8, 9, and 11 — the Learning Log, Build & Iteration, and Reflection — are written in the first person and dated, consistent with engineering-notebook practice. This split is intentional. The notebook sections record what was observed and decided at the time, including errors that were later corrected, and rewriting them in the third person would obscure the sequence in which the work was understood.

2. Executive Summary

This project set out to build a heart rate monitoring headband with Bluetooth connectivity, as an alternative to smartwatches for users whose preferences are not served by wrist-worn devices. The designer began with no prior experience in soldering, sewing, wiring, or laser fabrication, and completed a working prototype in eleven days.

The device was built around the materials and tools available. Every component was selected on the condition that it could be hand-soldered, laser-cut, or sewn. This constraint drove the selection of the GY-MAX30102 optical sensor over more capable but non-hand-solderable alternatives. The MAX30102 and an ADXL345 accelerometer are mounted at the temple in a laser-cut 3 mm black cast acrylic carrier, wired through the interior of the headband to a Seeed Studio XIAO ESP32C3 microcontroller, a 500 mAh LiPo battery, and a tactile button housed in a second acrylic module at the back of the head. The device recharges over USB-C, streams heart rate over standard Bluetooth Low Energy so that generic heart rate applications can connect to it, and pairs with a custom application that records timed activities and exports heart rate and motion data.

The prototype was tested against a Garmin Forerunner 255 as a commercial reference. At rest, it recorded a mean absolute error of 1.33 bpm. Recording while the user was running drove the mean absolute error to 14.65 bpm, approximately eleven times greater than at rest. The device operated for more than five hours on a single charge and maintained a Bluetooth connection to approximately 128 feet. Total device material cost was $52.07.

The device is accurate at rest and is not accurate in motion. The accelerometer transmits motion data, but that data is not applied to the heart rate calculation. Motion-artifact rejection was the accelerometer's stated purpose in the design and was not implemented; doing so is the most likely route to reducing the error observed under motion.

The primary outcome of this project is not the device but the fabrication capability behind it: design, cut, solder, sew, wire, and verify.

3. Vision & Purpose

The motivation for this project was to experience the engineering design process, learn new skills, and create a product that would be personally applicable. The design idea behind this project was to create a headband that would accurately monitor heart rate. The headband would serve as an alternative to smartwatches or rings based on the preference of the user.

User preferences could make a head-mounted heart rate monitoring system preferable to a wrist-worn watch for several reasons, including the following: the user prefers to keep long hair out of their face, or the user dislikes the comfort of wearing a watch.

Throughout this project, the designer set out to learn engineering skills, most notably fabrication and soldering.

4. Prior Art & Research

4.1 Commercial Landscape

The vast majority of heart rate monitoring devices feature bpm data as just one of numerous features. The most popular examples include the Apple Watch, Garmin watches, the Oura ring, and the Whoop band. Examples of dedicated heart rate monitoring systems include the Polar H10, which is widely considered the gold standard for heart rate monitoring, featuring a chest-mounted device that measures the electrical signals of the heartbeat, allowing for maximum accuracy.

Head-mounted heart rate monitors mostly include the Spree Bluetooth Fitness Headband and the Moov HR Sweat, which is the most similar. The Spree retailed for around $200, while the Moov HR Sweat retailed for around $100. However, both are discontinued products.

4.2 Honest Differentiation

This device competes only on user preference. It is neither the most accurate, the most versatile, nor the most affordable option available. Its distinguishing characteristic is form factor: the two closest commercial equivalents, the Spree Bluetooth Fitness Headband and the Moov HR Sweat, have both been discontinued, leaving the head-mounted category effectively unserved.

4.3 PPG Placement Literature

Khalili et al. compared photoplethysmography-based heart rate estimation across eight anatomical sites against electrocardiogram ground truth [1]. The study grouped sites by signal quality, reporting that measurements at the fingertips, finger base, earlobes, and forehead produced lower heart rate errors and stronger pulsatile signal components than measurements at the temple, mastoid bone, wrist, and forearm [1].

This finding has two implications for the present design. First, the temple is not among the strongest available sensing sites; the forehead, immediately adjacent and equally compatible with a headband form factor, is documented as producing better signal quality. The temple was selected in this project on the basis of comfort and retention rather than signal strength, and that trade-off was made without reference to this literature at the time.

Second, the reference device used for validation in §10 is wrist-worn, and the wrist is grouped with the temple among the weaker-signal sites. The comparison in §10 is therefore between two sites of comparable expected difficulty, rather than between a test site and a clinical standard.

The study further notes that motion sensors, commonly paired with photoplethysmography in wearable devices, can be used to capture motion-related artifacts and account for them in signal quality analysis [1]. This is the function the ADXL345 was included to serve in the present design, and its absence is discussed in §10.3 and §11.

5. Requirements & Constraints

5.1 Constraints

The following constraints were fixed at the outset of the project and shaped every subsequent design decision.

- Fabrication: all custom parts must be produced on a 10 W diode laser engraver, using 3 mm black cast acrylic. The machine cannot cut clear acrylic, as the beam passes through transparent material rather than being absorbed.
- Tooling: tools are limited to a soldering iron, sewing needle and thread, laser engraver, wire strippers, scissors, tweezers, and a multimeter. No 3D printer, oscilloscope, bench power supply, or scale was available.
- Component selection: all electronic components must be hand-solderable. Fine-pitch and reflow-only packages are excluded.
- Budget: device material cost under $100; total project expenditure under $140.
- Timeline: build and testing completed before the start of the school year.
- Skill: the designer began with no prior knowledge of soldering, sewing, laser fabrication, or small-scale electronics assembly.
- Verification: requirements were scoped to what the available instruments could measure. Quantities requiring equipment not owned, such as mass, were excluded from the requirements matrix rather than estimated.
- Testing population: single subject, single device. No variation in skin tone, head size, or fit pressure was tested.
- De-scoping: comfort optimization and cosmetic finish were explicitly excluded. The mechanical requirement is minimum viable retention during running.

5.2 Requirements Verification Matrix

ID	Requirement	Design Ref (§)	Verification Test	Result
R-01a	The device shall report average heart rate within 5 BPM of a reference device over a recorded activity while the wearer is stationary.	§7.1	§10.2	Pass
R-01b	The device shall report average heart rate within 5 BPM of a reference device over a recorded activity while the wearer is running.	§7.1	§10.3	Fail
R-02	The device shall remain in position during running without manual adjustment for 10 minutes.	§7.4	§10.6	Pass
R-03	The device shall operate on battery for at least 5 hours of continuous streaming.	§7.6	§10.4	Pass
R-04	The control module shall not exceed 1 inch in thickness.	§7.4	§10.7	Pass

ID	Requirement	Design Ref (§)	Verification Test	Result
R-05a	Device unit cost, comprising only components physically integrated into the finished prototype, shall not exceed $100.	§7.7	§7.7	Pass
R-05b	Total project expenditure, including tooling, spares, and consumed stock, shall not exceed $140.	§7.7	§7.7	Fail
R-06	The device shall stream heart rate to a phone over BLE with fewer than 2 disconnects per 20-minute session.	§7.5	§10.5.1	Pass

6. Concept Development

6.1 Alternatives Considered

A. Sensor selection

The following sensors were considered: the MAX30102, the MAXM86161, and the MAX86150. The MAXM86161 was rejected because it is not hand-solderable with the tools and skill available.

B. Enclosure approach

A 3D-printed hard enclosure, an off-the-shelf project box, and a fabric enclosure with acrylic mounting plates were considered. The 3D-printed option was unavailable, as no printer was owned, and fabric was judged better for a wearable regardless.

C. Sensor placement

Temple versus mid-forehead. The temple was selected on comfort and on the temporal artery's position.

D. Circuit design method

Browser-based circuit simulation versus a hand-drawn schematic were considered. The specific breakout boards were not in the simulators' part libraries, so a simulation would have modeled imaginary components, and it would have been more time-consuming to learn and use this software.

E. Power and charging

A separate TP4056 charging module versus the XIAO's onboard charging circuit was considered. The Seeed documentation confirmed the board charges a connected LiPo over USB-C [3], eliminating the need for the TP4056 charging module.

F. Extra features

Microspeakers were considered as an addition to the headband's functionality. This feature was not implemented on the basis of complexity. However, this addition could be implemented in v2.

6.2 Decision Matrices

Options were scored against weighted criteria on a scale of 1 (poor) to 5 (good).

Matrix 1 — Sensor selection

Selection Criteria	MAX30102	MAXM86161
Hand-solderable with available tools	5	1
Unit cost	4	3
Library and example code availability	4	2
Documentation quality	3	3

Selection Criteria	MAX30102	MAXM86161
Suitability for reflective temple PPG	4	4
Total	20	13

Matrix 2 — Enclosure approach

Selection Criteria	Fabric + acrylic backing	3D-printed enclosure	Project box
Producible with owned tools	5	1	3
Comfort against the head	4	3	3
Access for repair and iteration	2	4	4
Durability under flex	3	4	4
Bulk and weight	5	4	3
Total	19	16	17

6.3 Selected Concept

The selected concept places a MAX30102 optical heart rate sensor and an ADXL345 accelerometer on a shared I²C bus, temple-mounted in a laser-cut acrylic carrier, wired back to a Seeed Studio XIAO ESP32C3 in an acrylic control module sewn into an elastic headband, battery-powered and streaming over Bluetooth Low Energy.

The MAX30102 and ADXL345 were chosen for heart rate and motion data. They were chosen on the basis of hand-solderability with available tools. The acrylic mounting plates within a fabric enclosure were chosen on the basis of producibility with owned tools and low profile. The choices made supported affordability and the possibility to create and modify with available tools.

Factors conceded include the following: ease of repair and iteration, durability, additional features, and cleanliness of design. Overall, choices were made to maintain a simple project, making everything possible with the tools available.

7. Detailed Design

7.1 System Overview

The system contains the ADXL345 accelerometer and MAX30102 sensor attached to a 1 inch by 2 inch, 3 mm thick acrylic mounting plate. Both of these chips are wired to the Seeed Studio XIAO ESP32C3 microcontroller, which is also wired to a battery and a button, all of which are mounted to a second 3 mm thick acrylic plate measuring 1.1 inch by 3.9 inch.

7.2 Electrical Design & Pin Mapping

The device uses a seven-conductor run between the temple assembly and the control module. Both sensors are daisy-chained so that power, ground, and the two I²C signal lines are shared.

Wire	Signal	XIAO Pin	Destination
Red	Power	3V3	MAX30102 VIN → ADXL345 VCC
Black	GND	GND	MAX30102 GND → ADXL345 GND → button leg 1
Yellow	SDA	D4	MAX30102 SDA → ADXL345 SDA
Green	SCL	D5	MAX30102 SCL → ADXL345 SCL
Blue	INT	D2	MAX30102 INT
White	INT1	D3	ADXL345 INT1
White	Button	D6	Button leg 2
Green (link)	Jumper	—	ADXL345 VCC → CS

Note: the schematic drawing labels the D6 button connection as blue. In the assembled device, white wire was used for both the ADXL345 interrupt on D3 and the button on D6. The CS→VCC jumper on the ADXL345 is a short link internal to the sensor assembly and is not part of the seven-conductor run.

7.3 Sensor Configuration & I²C Addressing

The MAX30102 and the ADXL345 communicate with the Seeed Studio XIAO ESP32C3 over a shared I²C bus. Both sensors share the same two signal lines: SDA, the serial data line, on pin D4, and SCL, the serial clock line, on pin D5. Because each device on an I²C bus has a unique address, the microcontroller can address either sensor individually over the same pair of wires, and only the addressed device responds. The MAX30102 operates at address 0x57 [4] and the ADXL345 at address 0x53 [2].

The ADXL345 breakout supports both I²C and SPI communication. To force I²C mode, the CS pin is tied to VCC by a jumper. With the SDO pin left floating, the device defaults to address 0x53 [2]. This addressing scheme is what allows both sensors to be daisy-chained into a single shared run from the temple assembly back to the control module, rather than requiring a separate bus for each sensor.

7.4 Mechanical & Enclosure Design

The acrylic backing plates measure 1 inch by 2 inch for the temple-area mount and 1.1 inch by 3.9 inch for the control module. Both are cut from 3 mm black cast acrylic. Components are retained by friction fit rather than adhesive, allowing removal and replacement during iteration. The fabric enclosure method was chosen for optimal comfort and low profile.

7.5 Software & Firmware

The firmware and companion application were implemented by an AI coding assistant working to the author's specification. The description below is condensed from a technical summary written by that assistant; §11 states the division of authorship in full.

The firmware is a single file of roughly 430 lines running as one cooperative loop, with no operating system and no interrupts. On boot it initialises both sensors over I^2C and starts the Bluetooth stack; sensor failures are deliberately non-fatal, so a missing accelerometer cannot prevent heart rate from advertising. Heart rate is derived from beat-to-beat intervals, discarded unless between 20 and 255 BPM, and averaged over a four-sample buffer. That plausibility gate and four-sample mean are the entire filter chain: there is no band-pass filtering, no adaptive thresholding, and no outlier rejection. This is the direct cause of the motion error reported in §10.3.

The button is polled and debounced in software, recognising a single tap, a double tap, and a three-second hold. The firmware reports the raw gesture and assigns it no meaning; the application owns the session state, so the physical button and on-screen controls cannot diverge after a dropped connection. A three-second hold enters light sleep rather than deep sleep, because the button pin is not among the microcontroller's real-time-clock-capable pins and deep sleep could not have woken from it. This constraint arose from the pin assignment already being soldered and is discussed in §11.

The device advertises the standard Bluetooth Heart Rate Service, so generic third-party heart rate applications can connect to it, alongside a custom service carrying a seventeen-byte motion record — three accelerometer axes, computed magnitude, and flags for movement and reading confidence — and a one-byte button event. Data is transmitted once per second. Motion is sampled at 20 Hz but transmitted at 1 Hz, so the recorded value is a one-second snapshot rather than a peak or mean. There is no pairing or encryption; any device in range can read heart rate.

The companion application is plain HTML, CSS, and JavaScript with no framework or build step, served over HTTPS. This minimalism responds to a platform constraint: iOS Safari does not implement Web Bluetooth, so the application runs inside a third-party browser that provides no developer console, and a permanent on-screen debug panel is the only means of diagnosing a phone-side failure. The interface presents a dial, an activity timer, a per-second bar chart, and a connection indicator. Activities are stored on the phone and exported as comma-separated files with one row per second, carrying heart rate, all three accelerometer axes, magnitude, and the movement and confidence flags. This export produced the results in §10.

Known limitations:

- Motion data is recorded and exported but not acted upon. The confidence flag is transmitted and stored, but nothing consumes it and no reading is suppressed or corrected. Motion-artifact rejection, the original rationale for fitting the accelerometer, remains unimplemented.
- The first three readings after boot are mathematically incorrect, converging within four beats, and the rolling average never decays on loss of contact.
- The device carries no battery-sensing circuitry and cannot report its charge state.
- Several paths were verified in simulation only, including the standby and wake cycle and the reconnection fixes. There are no automated tests; all verification was manual.

7.6 Strain Relief & Durability

The main durability risk is associated with the soldered joints. Some joints appeared to be weak, but are now held in place with a large amount of hot glue, which should also protect the previously exposed joints from sweat and other contaminants. Additionally, the sewing on the headband is suboptimal, and the device is held together by a combination of lackluster sewing threads and electrical tape. However, it should be suitable for a prototype.

The device is recharged over USB-C through the microcontroller's onboard charging circuit, which charges the connected LiPo cell while also powering the board [3]. No separate charging module is required.

7.7 Bill of Materials

Item	Qty	Part No. / Spec	Source	Cost
MAX30102 Heart Rate and SpO2 Sensor Breakout Board	4	H1-HBB0203-2	hiBCTR (Amazon)	$5.50/unit
Tip Cleaning Kit	1	SWK-001	Wonderway (Amazon)	$19.11
DIYSELF Multimeter Tester	1	DIY1-SXWYB-001	DIYSELF (Amazon)	$12.99
DAOKI Micro Tactile Buttons	100	BG-US-91570	DAOKI (Amazon)	$0.05/unit
80W LCD Soldering Iron Kit	1	LJPXHHU-1	LJPXHHU (Amazon)	$17.98
Seeed Studio XIAO ESP32C3	2	102010633-FA	Seeed Studio Official (Amazon)	$10.99/unit
GY-291 ADXL345 3-Axis Digital Acceleration	1	3-01-0087-A	HiLetgo (Amazon)	$7.29
28 Gauge Wire Silicone Tinned Copper, 6 colors, 10 ft	1	mea230529ee0207	MECCANIXITY	$7.59
Magnetic Helping Hands, 10 Crocodile Clips	1	MM-USA-001-01	Siblaies (Amazon)	$9.99
8×12 inch 3 mm Thick Black Acrylic Sheet, 2 pieces	1	hhsyakeli0810	Lesnlok	$9.98
3.7 V LiPo Battery 500 mAh	1	601452	Liter Energy Battery (Amazon)	$9.99

Item	Qty	Part No. / Spec	Source	Cost
Black Headband	8	HF2505001	Yunaking (Amazon)	$0.675/unit
Total project expenditure (without tax)	—	—	—	$149.30
Cost of device materials	—	—	—	$52.07

8. Learning Log

Notebook — first person, dated. Entries were written at the end of each working day and are reproduced as recorded.

Entry #1

Tuesday, July 21, 2026 · Day 1 of active build

Goal for today

Get all the parts I received organized, then practice with the soldering iron and start getting comfortable with it before touching any real components.

What arrived vs. what did not

Eleven out of twelve parts came in today's shipment:

- Yunaking elastic headbands (8-pack)
- 80W adjustable soldering iron kit
- DIYSELF auto-ranging multimeter
- Magnetic helping hands (10 clips)
- Both Seeed XIAO ESP32-C3 boards (pre-soldered)
- HiLetgo GY-291 ADXL345 accelerometer
- hiBCTR MAX30102 heart rate sensors (2-pack)
- DAOKI tactile buttons (100-pack)
- Lesnlok 3mm black cast acrylic sheets (2 pieces)
- Liter energy 3.7V 500mAh LiPo battery (with PCM protection)

Still en route: MECCANIXITY 28 AWG 6-color silicone stranded wire.

Prior soldering experience

None. I've never soldered before today. I learned it entirely for this project, but I'm excited about picking up the skill. I believe I'll keep using it well beyond this build.

What I learned from the videos

I watched two YouTube videos on soldering before I touched the iron. Main takeaways that stuck with me:

- How to be safe while soldering
- Heat the joint first, then apply the solder to the joint (not the iron)
- Tin the tip of the iron with solder before working
- What a good joint should look like

What went wrong — the automotive wire

For practice, I started with an automotive wire I had around, likely 10-12 gauge. It's copper and much thicker than the 28 AWG I'll actually be using.

The wire wouldn't heat up enough for the solder to apply. It just sat on the surface without wetting into the strands. I took a photo of the failed joint.

I ran the iron through a range of temperatures trying to get it to work:

- Started at 360°C
- Bumped to 380°C
- Ended at 395°C

Even at the higher end, the solder wouldn't take properly. That was frustrating at the moment, but useful to see. However, because I got the iron so hot, I ruined one of the tips. It became black and oxidized, therefore it would not be able to transfer heat well.

How I'm feeling about it

I feel decent: excited, but not comfortable with soldering yet. I want to practice more before I touch the real parts. Honestly, I'm a little worried about messing up what I've bought. I don't want to have to reorder and spend more money, and I don't want to wait on shipping again.

What's next

- Hopefully the MECCANIXITY wire arrives in the next few days
- Once it's here, start visualizing the layout and measuring where each component sits
- Set up the coding environment — Arduino IDE, ESP32 board support, libraries

Entry #2

Wednesday, July 22, 2026 · Day 2

What I did today

I practiced more soldering, cut my first acrylic parts, and started sketching the physical layout. I also began to learn to hand sew.

Soldering practice Practiced on buttons instead of copper wire: thick copper turned out to be a bad practice medium. Soldering pin-to-pin at 350°C with a chisel tip felt way better. It took a few seconds per joint like the videos said, and the joints looked clean. Big confidence boost from yesterday.

I switched to a finer tip, planning to do real board work later, and kept practicing on button legs plus small wire strands to get it dialed in.

First acrylic cut

Cut a 1" × 3.5" piece of acrylic from the 12in×8in sheet. This will be the rigid backing plate for the electronics pod at the back of the head. It will hold the XIAO ESP32-C3, the battery, and the button. Test-placed it against the headband and the size feels right.

This was the biggest unlock of the day. My Falcon F1 10W handled the 3mm cast acrylic without issue. That confirms acrylic fabrication is genuinely available to me for this project, which opens up the entire mechanical side. This also opens up so many other projects.

Learning to sew

I also learned to sew today, for the first time. The headband is about 2.25 inches flat. I folded it over on itself to create a pocket, which is where all the electronics will sit. POST-EDIT: I changed this slightly, but in the final design I still had the wires in the middle of the fabric, and in some areas I did fold the headband fabric over itself and sew it, in order to halve the width.

One thing I had to account for: my thread doesn't stretch but the headband does. So I have to sew in extra slack, otherwise the stitching fights the elastic, and could break or rip out. The pod ends up a little over an inch thick, which isn't bad — though thinner would be better.

Layout sketch

Sketched the headband flattened out: where the sensor cluster goes at the front (temporal artery / forehead), how the wires route back, and where the electronics pod sits.

Real questions I worked through today

- Wiring. I was initially worried about needing 7+ wires per sensor. I2C lets both sensors share SCL, SDA, VIN and GND — so realistically only 5–6 wires run front-to-back, not 14. Big relief.
- Pin length. The pin headers are ~1/2" tall, too tall for a wearable. I'll trim them after soldering with flush cutters. POST-EDIT: I did not need to trim the pin headers, or at least did not attempt in order to not mess anything up, however this would still be a viable option for later models.

- Sweat. Sweat is conductive and will short exposed solder joints. Plan for v1: hot glue for insulation and strain relief at the sensor joints, electrical tape over larger exposed areas. Cleaner solutions (liquid electrical tape, conformal coating) are noted as v2 improvements.

Reflection Three completely new skills in one day: I got more comfortable with soldering, I cut acrylic on a laser for the first time, and I learned to sew. I'm learning a lot very fast right now.

What's next

- Read the Last Minute Engineers MAX30102 tutorial to solidify the I2C concept
- Continue design sketching
- Wait for the MECCANIXITY wire — that unlocks the rest

Entry #3

Thursday, July 23, 2026 · Day 3

Goal for today

Understand the circuit well enough to design it, then produce a complete wiring schematic I can build from — every wire, every pin, verified.

Reading and notes

Read both articles: the Last Minute Engineers MAX30102 tutorial and the SparkFun Learn I2C guide. I read the first one last night and took notes, then went back over both this morning.

1st article - lastminuteengineers

The MAX30102 is a sensor that uses both a red light LED and an infrared LED to monitor your heart rate and blood oxygen levels. The sensor works by shining both lights onto a thin part of your body and measuring how much light returns. The technical term is Photoplethysmography, which means measuring changes in blood volume using light. The infrared light is more absorbed by fresh oxygen rich blood, which occurs at each heart beat. Therefore, by determining the change in the wavelength being absorbed over time, the sensor can accurately measure your heart rate. The red light and the infrared light work in tandem to measure how well-oxygenated your blood is. Infrared light is absorbed more by oxygenated hemoglobin, and red light is absorbed more by deoxygenated hemoglobin; therefore, the sensor achieves the SpO2 level by comparing the different levels of absorption.

The FIFO buffer acts as temporary storage for the MAX30102 when sending data to the microcontroller.

My remaining curiosities or concerns aroused by this article include the following: will the sensor be accurate when placed on the temple area, how does SpO2 help you determine your performance, what are the basic details of how the signal is actually sent between the parts, and how can I make sure the sensor is pressed against the optimal area in order to accurately read heart rate.

2nd article - SparkFun Learn I2C is able to support multiple "peripheral" digital integrated circuits to communicate with one or more "controller" chips. Each I2C bus consists of two signals: SDA and SCL. SDA is the data signal and SCL is the clock signal.

The part that clicked afterward, and that makes the whole wiring plan make sense: every device on the bus has its own address. The MAX30102 is at 0x57 and the ADXL345 at 0x53. The XIAO addresses one, sends on SDA, and only that device answers. That's why two sensors can share the same two wires — and it's why I only need four shared lines plus interrupts instead of a separate bus per sensor.

Open research questions

These came out of the reading, and they're the questions I actually want answered by the end of this project:

1. Will the sensor be accurately placed on the temple? 2. How does SpO2 help determine performance? 3. What are the basic details of how the signal is physically sent between the parts? 4. How do I make sure the sensor is pressed against the optimal area to read heart rate accurately?

Question 1 is the big one and it's the thing the testing phase has to answer.

Digital circuit design: investigated, then rejected

My plan for today was to design the whole circuit and system digitally, then replicate it in real life. I looked into browser-based tools for this.

I ended up not doing it for the purpose of simplicity. The simulator also likely wouldn't have the exact parts I wanted.

Decision: hand-draw the schematic

On sourcing

I decided to attempt to pull only from published and well regarded sources. This decision is because it is both more reliable and it is what belongs in an engineering report. Every pin assignment on the schematic got checked against the manufacturer documentation.

The wiring schematic

Drew the full schematic by hand. Seven wires, six colors, from ten-foot spools. Both sensors daisy-chained into a single cluster, with one run back to the XIAO.

Things I worked out while drawing it:

- Ground daisy-chain order doesn't matter. I'd wired it in a different physical order than the plan and wanted to check. Whichever routing is easier for the headband layout is the correct one.

- My board is the GY-MAX30102 with 5 holes, though the header strip has 6 positions. The 5 populated ones — VIN, GND, SCL, SDA, INT — are all I need. POST-EDIT: The second order of GY-MAX30102 that arrived in shipping only had 5 header pins in the strip.

- The BAT+ and BAT− pads are on the underside of the XIAO, and they're flat surface-mount pads, not through-holes. No hole for a wire to poke through, so nothing holds the wire mechanically while I solder. The technique is different: tin the pad, tin the wire, then hold them together and touch the iron to both at once so the two beads merge. Because there's no mechanical support, hot glue over the finished joint isn't optional — one flex and the joint tears off the pad, potentially pulling copper trace with it and killing the board.

- The XIAO charges the LiPo over USB-C by itself. It has an onboard charging circuit, so it powers itself and charges the connected battery at the same time, and runs off the battery when unplugged. This made a separate TP4056 charging module unnecessary.

Acrylic Board Mount Design

In order to fabricate the acrylic board to mount the hardware, I had to learn how to use Inkscape for my designing tool and lightburn to control my laser cutting settings. I did research on what settings are ideal for acrylic, and then after one or two changes, saved those settings to my material library. I went through three variations of the Seeed XIAO mount individually, both precise measurement and trial and error to determine a friction fit, then another four variations of the entire backplate for that would hold the Seeed XIAO, battery, and button changing hole size for the battery and button and adding 1-2mm holes for sewing thread through to attach to the headband.

What's next

- Finalize and cut the temple sensor mount
- Test that the XIAO actually runs code
- Start soldering headers onto the MAX30102 once the wire arrives

Entry #4

Friday, July 24, 2026 · Day 4

Goal for today

Finalize and cut the temple-area sensor mount, verify the XIAO actually runs code, and start soldering headers onto the MAX30102 sensors.

Temple mount — likely final version Printed off what is the likely final copy of the temple-area mount, which is the acrylic piece that holds the MAX30102 and the ADXL345 at the right place in the headband. The design required more work in Inkscape and LightBurn. POST-EDIT: This variation was not the final copy, as I had to change the friction fit slot for the MAX30102, as the new one I ordered had the matching five header pins instead of six. I also added more holes for the sewing.

XIAO functional test — passed

Plugged the XIAO into my laptop, downloaded the build, and watched the LED blink on and off in the expected pattern in the terminal (no actual LED light was blinking, just feedback in terminal). The board runs code. That is confirmed before anything gets wired to it.

First real wire splice

Practiced with smaller wire, around 28 AWG, and tinned it reasonably well. Made an actual wire-to-wire connection for the first time. It would likely be considered a terrible connection by experienced solderers, yet it is a step forward.

Soldering the MAX30102 headers

This process was roughly three hours of trial and error.

The first sensor failed.

I got it connected but flooded it with solder. Three pins in a row bridged. While cleaning that up and reworking it, two things went wrong: I pushed three of the pins down too far because the board got so hot from too much time under the iron, and at some point the iron tip touched the LED/sensor window and left a bubble or crack. Between the pushed pins and the damaged window, I'm treating this one as scrap.

Second sensor was possibly usable. With the lesson learned, the second one came out functional. Not pretty, likely more solder than there should be on each pin, and two pins look like they might be joined by a tiny bridge in the middle. However, the pins are shiny and mechanically locked. I can't pull them out. It doesn't look like the videos, but it's holding.

What I learned — the tip/temperature progression

The real lesson was finding the setup that actually works on my hardware. I worked through it in order:

1. Skinny tip at 350°C — wouldn't work 2. Skinny tip + rosin-core flux — still wouldn't work 3. Big tip (more surface area) + lots of flux — worked noticeably better, but the tip got dirty fast and I don't think

it was holding heat 4. Chisel tip at ~360–370°C + flux, going slower — this is what finally worked Working theory on root cause: I suspect the iron itself is part of the problem. It's the cheap one from a ~$17 kit, and I think the tip temperature may be fluctuating rather than holding the setpoint. I'm sure I'm still doing things wrong too, but I'm following the video technique closely and the heat delivery feels inconsistent.

Ordering / spares

I already ordered one more 2-pack of MAX30102 sensors. I may order another ADXL345 as a spare before I attempt soldering it, since I haven't touched the accelerometer yet. POST-EDIT: I never ordered another ADXL345.

How I am feeling

I am drained, mentally and physically. Three hours of close-focus soldering drained my energy. I wanted to keep pushing but I could tell I was done for the day. Still, real progress: the XIAO runs, I made my first wire splice, and I have one usable MAX30102.

What's next

- Next session: attempt ADXL345 headers, then the CS→VCC jumper, then daisy-chain the sensors

Saturday 7/25 – Monday 7/27: Absent

Entry #5

Tuesday, July 28, 2026 · Day 5

Goal for today

Get back on the build after a break. Redesign the temple sensor mount for the new 5-pin MAX30102, cut the wire runs, and solder the complete harness. Make every connection except the battery.

Warm-up: practicing on the new wire

Started by practicing wire-to-wire splices with the MECCANIXITY 28 AWG silicone-stranded wire that had finally arrived. The splices came out rough-looking, but noticeably easier than my first attempts last week.

I practiced wire-to-pin joints on the scrapped MAX30102 from Friday. I soldered wires onto all six of its pins as to gain experience. That board was already scrap, so it cost me nothing to learn on.

Soldering the real sensors With the practice done, I soldered both new MAX30102 boards. The joints actually looked like the ones in the reference video. After Friday, that felt like a breakthrough.

I only had the one accelerometer on hand and no spare ordered, so I went in nervous. However, it came out nearly perfect. Likely the cleanest work I've done so far.

The wiring schematic

I drew the wiring schematic semi-to-scale. I drew the acrylic pieces to actual scale and set them about 12 inches apart on the paper, then planned for 16 inches of wire per run from the back module to the temple. The extra length is deliberate: the elastic headband stretches and the wire doesn't, so the wire needs enough slack that the elastic never puts tension on a solder joint.

Redesigning the temple mount

The new MAX30102 boards have 5 pins. The older ones had 6, as there was a stray pin sitting past the 5 holes. So the slot in the temple acrylic mount had to be redesigned to match the new size.

The harness, three hours straight

Cut the six 16-inch runs and soldered the complete harness: power (3V3), ground, SDA, SCL, both interrupt lines, and the button leg. I soldered everything to the Seeed XIAO first, then connected each end of the wire with a smaller one, in order to daisy-chain the two sensors together. I did this by pre-tinning multiple wires into a single shared pin joint. The only breaks were for mid-build continuity checks with the multimeter.

Testing

Continuity-tested the whole harness in continuity mode with everything unpowered. I went through every configuration: each daisy-chained line, each direct run to the MAX30102, and the interrupt lines that weren't part of any chain. Every connection beeped where it should, and did not beep where it shouldn’t.

Decisions and things I abandoned

- Wrapping wire around each pin: abandoned. The pins are far too small to wrap. Switched to lap-soldering the wire flat along the long side of the pin.
- Snipping the pins: abandoned. Confirmed that lap-soldering to the full pin is a normal practice.
- Heat-shrink: abandoned. Did not have any heat shrink small enough, and decided hot glue would serve the purpose better.
- Soldering pins while the board sits in the acrylic: rejected. My iron runs at ~360°C and cast acrylic deforms far below that; the heat would conduct into the slot and wreck the mount I just cut. Solder on the bench, friction-fit the finished assembly after connections are made. However, I had to therefore be careful about threading the wire through the acrylic piece before making any connections.
- Headband comfort and aesthetics: explicitly de-scoped. The requirement is minimum viable retention: it has to grip my temple and stay in place during a run. Plush comfort and product-level polish are out of scope because of the timeline, and because this is a personal prototype, not a product.

New skills and gear

- Added a new iron tip and a tip cleaner mid-day. Noticeably better control afterward.
- Pre-tinning both surfaces then joining. This solved a major problem: trying to hold two tiny parts together in mid-air while also holding the iron and the solder. Tin the pin, tin the wire, then just reheat them together.
- Continuity testing on a multimeter. I didn't know how to use the multimeter at all before today. The thing that unlocked it: continuity mode needs no power in the circuit, as the meter sends its own small current down one probe and listens for it on the other. You verify the joints before you ever apply 3.3 V, so you never push power into a short.

How I'm feeling about it

Big confidence jump. Eight days ago I had never soldered anything. Today I built a complete wired harness and every continuity check landed where it should.

Honest assessment though: the soldering is functional, not good. It's beginner-level work. The joints may fatigue or fail over time with the headband flexing, and that's a real known risk, not something I want to paper over. POST-EDIT: the soldered joints are my biggest concern still, however the large amount of hot glue around any joint should provide enough structural support and protection from sweat or other contaminants.

What's next

1. Hot glue each tested joint for strain relief 2. Electrical tape over the finished bundle 3. Test on the laptop: run the I2C scanner, expecting 0x57 and 0x53 4. Battery last, only once everything is verified 5. Then headband integration

Entry #6

Wednesday, July 29, 2026 · Day 6

Goal for today

Power the harness up for the first time, confirm both sensors are talking, then get the battery wired and the electronics into the headband.

First power-on: the I2C scan Plugged the XIAO into my laptop and ran the scan through PlatformIO in the terminal. Both addresses came back immediately: 0x57 for the MAX30102 and 0x53 for the ADXL345. No troubleshooting, no retries.

Then I uploaded to the device and monitored feedback in the terminal on each of the test sketches in turn. Every one returned positive results. I was getting live motion feedback from the accelerometer and live heart rate feedback from the pulse sensor. Everything I'd soldered on Tuesday was connected and working.

The blue wire came off

At some point after that, the blue wire, the MAX30102 interrupt line running to D2, broke off the pin. I'm not sure exactly how or when. I re-soldered it and got it back on.

Soldering the battery terminals: three hours, and the hardest thing I've done

This was by far the most difficult and most stressful part of the entire project.

The battery pads are tiny. They are much smaller than any pin I'd worked on. And the whole time, I had to keep the positive and negative leads from touching each other, on a live cell, with an iron in my hand. That was a different kind of pressure than working on an unpowered board. Holding the small wire in place while trying to solder, even with the pads and wire pre-tinned, was nearly impossible. I attached the wires to both terminals at least twice, and then one connection would fail and break.

The wires did touch, several times. Sparks flew, and I was concerned about the battery shorting, but I continued to check the voltage with the multimeter to confirm it wasn’t shorted.

Three hours in, I finally got both terminals soldered.

Confirming the battery works

Once everything was set in place, I tested it again. The heart rate sensor's LED came on while the XIAO was unplugged from my laptop, meaning that the board was drawing power from somewhere, and the only somewhere was the battery. The battery was wired correctly.

Sewing: it became wearable

After another break I moved to the headband. I cut the openings I needed in the fabric and sewed it together around the electronics, using the inside of the headband to conceal the wires.

By the end of the night the headband was wearable. I put it on and everything sat where it was supposed to. That's the first time this thing existed as an actual object you wear rather than a pile of connected parts on a bench.

Software

I also did work with Claude Code today. Documenting that separately: the firmware and app side needs its own write-up rather than a line in a build log. What's next

- Build out the phone application and get Bluetooth working
- Full functional test on battery power alone
- Heart rate accuracy testing against a reference
- Motion testing

Entry #7

Thursday, July 30 – Friday, July 31, 2026 · Days 7–8 (Combined for sake of continuity)

Summary

I built the full phone application, established the BLE connection to the headband, and continued troubleshooting it to work. Activity can now be viewed, logged, and exported on the phone.

Phone application

The software is a live website hosted on vercel and used through the Bluefy browser, enabling constant bluetooth connection with the headband. The website went through several iterations including UI and sensor readings.

Connection Dropout Issues

I began wearing the device and using it with the app for periods around 5 to 10 minutes, in which bluetooth connection would drop off rapidly going between connected and disconnected every few seconds, mainly ending in disconnection.

Accelerometer

At this point in the build, the accelerometer was completely dead weight. No motion data was being transmitted. I made the decision to allow for the motion data to be included on the data export of an activity, including x, y, z data and the acceleration data.

AI Tooling

Claude Code was used heavily across both days to build and debug the firmware and the companion application. I used it as a learning tool: it worked through implementation problems, and produced a written technical summary of what was built, which I read and then used as the basis for the software description in §7.5. The Learning Log entries in this section are my own, written at the end of each working day.

9. Build & Iteration

Notebook — first person. This section reorganizes the build record by subsystem rather than by date.

9.1 Soldering Progression

I had never soldered before starting this project, nor did I know anything about it. I learned it entirely for this project. I watched two YouTube videos on soldering before I touched the iron [5], [6]. I learned to heat the joint first, then apply the solder to the joint rather than to the iron, to tin the tip of the iron with solder before working, and what a good joint should look like.

For practice, I started with an automotive wire I had around, likely 10–12 gauge. It is copper and much thicker than the 28 AWG I would actually be using. The wire wouldn't heat up enough for the solder to apply. It just sat on the surface without wetting into the strands as it had in the reference videos. (See Figure 2). I ran the iron through a range of temperatures trying to get it to work: started at 360 °C, bumped to 380 °C, and ended at 395 °C. Even at the higher end, the solder wouldn't take properly. That was frustrating at the moment, but useful to see. However, because I got the iron so hot, I ruined one of the tips. It became black and oxidized, and therefore would not be able to transfer heat well. (See Figure 3.)

I practiced on buttons instead of copper wire. Soldering pin-to-pin at 350 °C with a chisel tip felt much better. It took a few seconds per joint like the videos said, and the joints looked clean. This was a big confidence boost. I switched to a finer tip, planning to do real board work later, and kept practicing on button legs plus small wire strands to get it dialed in. I practiced with smaller wire, around 28 AWG, and tinned it reasonably well. I made an actual wire-to-wire connection for the first time.

The MAX30102 headers, the first real components, took roughly three hours of trial and error to solder, and overall ended unsuccessfully the first day. I got the pins connected to the joints but flooded the area with solder. Three pins in a row bridged. While cleaning that up and reworking it, two things went wrong: I pushed three of the pins down too far because the board got so hot from too much time under the iron, and at some point the iron tip touched the LED sensor window and left a bubble or crack. Between the pushed pins and the damaged window, I treated this one as scrap. (See Figure 4.) With the lesson learned, the second one came out functional. (See Figure 5). Not pretty, likely more solder than there should be on each pin, and two pins look like they might be joined by a tiny bridge in the middle. However, the pins are shiny and mechanically locked. I couldn't pull them out. It did not look like the videos, but it held.

The real lesson was finding the setup that actually worked on my hardware. I worked through it in order: skinny tip at 350 °C, which wouldn't work; skinny tip with rosin-core flux, which still wouldn't work; a big tip with more surface area and plenty of flux, which worked noticeably better but got dirty fast and I don't think was holding heat; and finally a chisel tip at approximately 360–370 °C with flux, going slower, which is what finally worked. My working theory on root cause is that the iron itself was part of the problem. It is a cheap one from a roughly $17 kit, and I think the tip temperature may have been fluctuating rather than holding the setpoint. I added a new iron tip and a tip cleaner after a three-day break. There was noticeably better control afterward.

I practiced wire-to-wire splices with the MECCANIXITY 28 AWG silicone-stranded wire that had finally arrived. The splices came out rough-looking, but noticeably easier than my first attempts. I practiced wire-to-pin joints on the scrapped MAX30102 from Friday. I soldered wires onto all six of its pins in order to gain experience. That board was already scrap, so it cost me nothing to learn on. With the practice done, I soldered both new MAX30102 boards. The joints actually looked like the ones in the reference video. This felt like a huge breakthrough. I only had the one accelerometer on hand and I hadn't ordered a spare, so I went in nervous. However, it came out nearly perfect. I did most likely the cleanest work on the accelerometer. (See Figure 6.)

Soldering the battery terminals was by far the most difficult and most stressful part of the entire project. The battery pads are tiny. They are much smaller than any pin I had worked on. And the whole time, I had to keep the positive and negative leads from touching each other, on a live cell, with an iron in my hand. That was a different kind of pressure than working on an unpowered board. Holding the small wire in place while trying to solder, even with the pads and wire pre-tinned, was nearly impossible. I attached the wires to both terminals at least twice, and then one connection would fail and break. Additionally, the wires did touch, several times. Sparks flew, and I was concerned about the battery shorting, but I continued to check the voltage with the multimeter to confirm it wasn't shorted. Three hours in, I finally got both terminals soldered. (See Figure 7.)

At some point after that, the blue wire, the MAX30102 interrupt running to D2, broke off the pin. I'm not sure exactly how or when. I re-soldered it and got it back on.

My honest assessment is that the soldering is functional, not good. It is beginner-level work. The joints may fatigue or fail over time with the headband flexing, and that is a real known risk. The soldered joints are my biggest concern; however, the large amount of hot glue around each joint should provide enough structural support and protection from sweat and other contaminants.

9.2 Enclosure Iterations

I went through several iterations of both acrylic backing boards. At first, I cut out a 1 inch by 3.5 inch piece from the acrylic to test out my laser engraver, and then used this board to test several friction-fit mounts for the Seeed Studio XIAO. I went through two trials for the right friction fit, and then another to allow the battery terminals on the bottom to be exposed.

After solidifying the spacing for the microcontroller, I moved on to designing spacing for the battery and the button, going through several iterations, including the addition of 1 mm holes around the board for sewing, and then finally landing on the final cutout. (See Figures 8 through 12.)

Next, I moved on to designing the temple-area mount for the ADXL345 and the MAX30102. I went through several friction-fit alternatives, having to adjust for the raised pieces on the bottom of the MAX30102, and settled on a design for the 6-pin variant that I had originally received. I added 1 mm holes for sewing and switched the spacing, so that the long side of the pins would be farther apart to make wiring easier. After I received the 5-pin version I had to redesign the friction-fit slot of the MAX30102, but kept everything else the same.

9.3 Sewing & Fabric Assembly

I began with a small demo by folding the headband together end to end and sewing the ends together. After that, I began the actual fabric housing by cutting two holes, spaced apart properly for the temple and controller areas, and threaded the assembly through. I sewed a few threads through the holes to keep the mounting plates in place, and later folded some of the sections and went around, halving the width of the headband. The temple mount with the MAX30102 is on the inside of the headband, while the control module is on the outside, with the wires concealed inside the fabric. (See Figures 13 and 14.)

9.4 Harness Assembly

I drew the wiring assembly schematic by hand. The first schematic I drew to get an idea of the wiring order, and the second I drew roughly to scale. (See Figures 15 through 17.) I learned about the purpose of each wire connection. All of the attaching is described in the soldering progression section. However, after attaching the wires, I had to verify that the soldered connections were good and that no joints were bridged together. I used the multimeter continuity test to verify that each joint was properly connected. Continuity mode requires no power in the circuit: the meter sends its own small current down one probe and listens for it on the other, so the joints are verified before 3.3 V is ever applied.

10. Testing & Results

10.1 I²C Bus Scan

The I²C scan returned both device addresses, 0x53 and 0x57, confirming that the I²C signal, power, and ground connections to both sensors were electrically sound. (See Figure 18.)

10.2 HR Accuracy vs. Reference (Resting)

Method

Three simultaneous two-minute recordings were taken at rest, seated and stationary, on August 2, 2026. The prototype logged heart rate once per second to the companion application; the reference device recorded the same interval using its Health Snapshot function. Both devices were started within one second of each other, confirmed by comparing the prototype's first sample timestamp against the reference device's recorded start time.

Reference device: Garmin Forerunner 255, Health Snapshot, two-minute protocol. The reference is a wrist-worn optical PPG device, not an electrocardiographic instrument. Because it shares the same underlying sensing technique as the prototype, the comparison measures agreement with a consumer reference, not accuracy against clinical ground truth. Chest-strap ECG would be required for the latter.

Conditions: the prototype's accelerometer recorded a mean magnitude of 9.66–9.89 m/s^2 across all three trials, with the moving flag false for 100% of samples (352 of 352). The onboard confidence flag reported valid contact for 100% of samples. All three trials therefore represent genuinely stationary, good-contact conditions.

Results

Trial	Start	Duration	Samples	Prototype avg	Reference avg	Difference	Ratio
1	3:14 PM	119 s	117	53 bpm	54 bpm	−1	0.981
2	3:30 PM	122 s	119	56 bpm	58 bpm	−2	0.966
3	3:34 PM	120 s	116	58 bpm	57 bpm	+1	1.018

Mean absolute error: 1.33 bpm. Bias: −0.67 bpm, indicating the prototype reads marginally low. Mean ratio: 0.988.

Peak value comparison

Trial	Prototype max	Reference max	Difference
1	62 bpm	60 bpm	+2
2	64 bpm	72 bpm	−8
3	71 bpm	63 bpm	+8

Mean absolute error on peak values: 6.0 bpm, roughly four and a half times the error on the averages.

This divergence is expected and informative. An average over 120 samples suppresses uncorrelated noise, whereas a maximum is a single-sample extreme and therefore fully exposed to it. The prototype applies only a four-sample rolling mean and a plausibility window of 20–255 bpm, with no outlier rejection or signal-quality weighting, so isolated spurious beats propagate directly into the reported maximum. The result indicates that the prototype tracks central tendency well under stationary conditions but is not reliable for instantaneous or peak readings.

Interpretation

Agreement on mean heart rate is within approximately 1–2 bpm across all three trials, satisfying R-01a. The ratio of approximately 0.99 confirms that the prototype is tracking the true pulse rate rather than a harmonic of it; an alternate-beat detection failure, which had been considered as a candidate explanation for earlier low readings, would have produced a ratio near 0.5. No such pattern is present.

This result answers the first open research question recorded in Learning Log Entry #3 — whether the sensor would read accurately at the temple — in the affirmative, for stationary conditions only. Performance under motion is addressed separately in §10.3. (See Figures 19 through 21.)

Limitations

1. The comparison uses summary statistics, not paired samples. The reference device's Health Snapshot exports only aggregate values — average, maximum, and HRV — rather than a per-second series. Statistical power is therefore n = 3 paired averages, not n = 352 paired samples. A per-second comparison would require exporting a recorded activity in a format containing the full time series.
2. Single subject, single session. All trials were recorded on one wearer within a twenty-minute window. Sensor performance is known to vary with skin tone, perfusion, and fit pressure, none of which were varied here.
3. The reference is not ground truth, as stated above.
4. Rest only. These trials establish nothing about performance under motion.

10.3 HR Accuracy vs. Reference (Motion)

Method

Two simultaneous recordings were taken under motion on August 3, 2026. The prototype logged heart rate once per second to the companion application; the reference device recorded the same intervals as standard running activities, exporting a full per-second time series.

Reference device: Garmin Forerunner 255, running activity, 1 Hz recording. As in §10.2, the reference is a wrist-worn optical PPG device and not an electrocardiographic instrument.

Trial 1, 8:54 AM. Duration 10 min 10 s at a constant pace of 6.0 mph (9.7 km/h), equivalent to a ten-minute mile. The reference device logged 609 samples over 608 s.

Trial 2, 9:19 AM. Duration 3 min 1 s using a progressive protocol: walking at the start, increasing to a running pace and continuing to increase in speed over the course of the trial. Exact speed transitions were not recorded. The reference device logged 181 samples over 180 s.

Both trials were conducted on a treadmill. Trial 1 was recorded using the reference device's standard outdoor running mode rather than its treadmill mode, so the recorded distance of 102 m reflects satellite positioning of a stationary runner and is not meaningful. Trial 2's recorded distance of 548 m at 10.9 km/h derives from wrist-based estimation. Neither distance figure is used in this analysis.

Synchronisation was confirmed by timestamp: the prototype's session labels of 8:54 AM and 9:19 AM correspond to reference activity start times of 13:54:41 Z and 14:19:28 Z respectively, with recorded durations agreeing to within two seconds in both trials.

Results

Trial	Condition	Duration	Prototype avg	Reference avg	Difference	Ratio
1	Treadmill, constant 6.0 mph	10:10	142 bpm	126.1 bpm	+15.9	1.126
2	Treadmill, progressive	3:01	140 bpm	126.6 bpm	+13.4	1.106

Mean absolute error under motion: 14.65 bpm. By comparison, the mean absolute error at rest was 1.33 bpm. Error under motion is approximately eleven times greater.

Extremes

Trial	Prototype peak	Reference peak	Prototype low	Reference low	Prototype range	Reference range
1	177	142	74	68	103	74
2	180	159	55	85	125	74

The prototype reported a minimum of 55 bpm during Trial 2 and the reference reported 85 bpm over the same interval. Both minima occurred during the initial walking phase, before running speed was reached. The prototype's value is consistent with the wearer's measured resting heart rate of 53–58 bpm (§10.2), while the reference's value is higher than the wearer's resting rate. Neither device can be independently verified at this point in the trial, and the discrepancy is not resolvable from the available data: the prototype may have been tracking a genuinely low pre-exertion heart rate, or the reference may have been responding to early movement. This trial therefore does not establish a dropout failure, and no such claim is made. (See Figures 22 and 23.)

Interpretation

The prototype does not measure heart rate accurately under motion. It over-reported mean heart rate by 13–16 bpm in both trials, in the same direction and of similar magnitude. This is consistent with mechanical motion — footstrike impact and headband movement against the skin — producing periodic optical fluctuations that the beat-detection algorithm counts as pulses. Spurious detections raise the reported rate.

The direction of the bias reverses between conditions: at rest the prototype read marginally low, at −0.67 bpm, while under motion it reads substantially high, at +14.65 bpm. This confirms the errors are motion-induced rather than a fixed calibration offset.

This result is consistent with the placement literature reviewed in §4.3 and with the design limitations documented in §7.5. The prototype's signal processing consists of a four-sample rolling mean and a plausibility window of 20–255 bpm. It applies no band-pass filtering, no adaptive thresholding, and no motion-artifact rejection. Critically, although the ADXL345 accelerometer was included in the design for the stated purpose of motion-artifact rejection, that capability was never implemented: motion data is recorded and exported alongside heart rate, but does not influence the reported value.

Conclusion

This answers the open research question first recorded in Learning Log Entry #3 — whether the sensor would read accurately at the temple — with a qualified negative. The prototype achieves close agreement with a consumer reference at rest, but does not achieve usable accuracy during running, which is the condition the device was primarily intended for. R-01a is met; R-01b is not.

The failure is attributable to the absence of motion-artifact rejection rather than to sensor placement alone, and is addressed as the leading item in §11.

Limitations

5. Summary statistics rather than paired samples. Prototype values were read from the application's activity summary. A per-second alignment of both time series would permit calculation of correlation and of error as a function of measured acceleration; the exported data supports this analysis, which was not performed within the project timeline.
6. Two trials, one subject, one session.
7. The reference is itself an optical PPG device and is subject to its own motion artifacts, though it benefits from a wrist form factor with substantially better mechanical coupling.

10.4 Battery Runtime

The device operated continuously for more than five hours on a single charge. Beyond approximately five hours, standby entry failed to trigger and heart rate readings degraded; the device was powered down shortly after, so total runtime to depletion was not established. The device carries no battery-sensing circuitry, so the terminal voltage at the point of degradation could not be measured. Adding a voltage divider to an analog input is identified in §11 as a v2 improvement.

10.5 Bluetooth Reliability

10.5.1 Timed connection test

Debug log, August 1, 2026:

```
15:51:38  connected to HR Headband
15:51:38  screen wake lock acquired
15:52:33  screen wake lock released
15:56:02  button event code 1
15:56:03  activity started
16:18:07  button event code 1
16:18:07  activity paused at 22:05
16:18:12  button event code 2
16:18:12  activity ended: 22:05, 1308 samples
16:18:12  saved activity 1785617762906
```

From the data, the headband was connected to the application for over 25 minutes, maintaining connection through an activity lasting 22 minutes 5 seconds with 1,308 samples recorded and no disconnections. This test was conducted at rest, with both the headband and the phone stationary and within 2 feet of each other at all times. R-06 is satisfied.

Both motion trials in §10.3 also completed without connection loss, at 10 min 10 s and 3 min 1 s respectively, while the wearer was running and the handset was carried on-person.

10.5.2 Maximum range

The device lost Bluetooth connection at roughly 128 feet, measured outdoors with line of sight.

10.6 Retention Test

The device remained in place with the sensor on the correct area of the temple throughout a 10-minute run, with no manual adjustment required. R-02 is satisfied.

10.7 Thickness and Wearability Test

The device does not exceed 1 inch in thickness at any point. The device meets general comfort expectations and maintains a relatively low profile. R-04 is satisfied. (See Figures 24 through 27.)

11. Reflection & Future Work

Notebook — first person.

I believe that the project achieved what I largely set out to accomplish. While I solved a practical issue by creating the headband, the main goal of this engineering project was to learn. I learned a multitude of skills including, but not limited to, fabricating acrylic with a laser engraver, soldering, sewing, and wiring. I also learned an incredible amount about electronics and how the microchips interact with each other, and I learned about optical heart rate sensors and how photoplethysmography works.

If I redid this project specifically, there are a few things I would do differently for v2. The first main thing is the accuracy of the device in motion. From the presented data, I would attempt to solve the accuracy issue by cancelling out the motion disturbance using data from the accelerometer. The data from the accelerometer is exported; however, it is not applied in the application. I would attempt to integrate small speakers into the headband and allow it to stream music. I would also build the application to a higher level and have it track SpO2 data, leveraging as much information as I could get from the MAX30102 sensor, and possibly add one or two more sensors that could monitor more aspects of health, such as respiration. This would allow the application to provide more information for overall health. Additionally, I would use a higher-quality headband to add more protection to the device. I would also add battery-sensing circuitry so the device can report its own charge state, which the current build cannot do. For future work, this project has enabled me to attempt more electronic and soldering projects and to utilize my ability to fabricate acrylic with my laser engraver.

On the Use of AI Tooling

This project was built with the assistance of AI tools, and this section states plainly what they did and did not do, so that no claim made elsewhere in this report is ambiguous.

The physical device is entirely my own work. Every solder joint on this device was made by me, including the two battery terminal pads that took three hours. I diagnosed the soldering problems myself: the oxidized tip, the thermal mass of the practice wire, and the tip-and-temperature progression that finally produced usable joints. I designed every acrylic part in Inkscape, determined the laser settings in LightBurn, and iterated the backplate through seven variations to reach a friction fit. I learned to sew for this project and assembled the headband by hand. I drew the wiring schematic, verified every pin against manufacturer documentation, and continuity-tested the complete harness before applying power. When a joint failed in service, I found it and repaired it.

The design decisions about the physical system are also my own. The choice of the MAX30102 over the MAXM86161, on the grounds that the latter cannot be hand-soldered with the tools available to me. The decision to build a fabric-and-acrylic assembly rather than a 3D-printed enclosure. Temple placement. The rejection of circuit simulation in favor of a hand-drawn schematic verified against datasheets, because the specific breakout boards I own do not exist in simulator part libraries. Sixteen inches of wire per run, so that the elastic never loads a solder joint. The decision to solder on the bench and friction-fit afterward, because my iron runs hotter than the temperature at which cast acrylic deforms.

The testing is my own. I designed the comparison protocol, ran the trials, and interpreted the results, including the finding that the device fails under motion.

The firmware and companion application were written by an AI coding assistant, Claude Code, working under my direction. I specified what the application should do and how it should behave: the gauge geometry and its range, the control layout, the button gesture vocabulary, the color semantics, and the requirement for data export. Those product and interaction decisions were mine and were specified in enough detail to be implemented directly. The engineering decisions underneath them were not mine. The choice of BLE stack, the sensor libraries, the Bluetooth service structure, the storage model, and the overall application architecture were proposed by the assistant and I accepted them, in most cases without challenge. The one exception is documented: when the assistant proposed relocating the button to a different pin to enable deep sleep, I refused, because the wiring was already soldered. That refusal forced a light-sleep implementation instead, which costs roughly 150 µA against 5 µA in standby — a trade I made deliberately to preserve completed work.

The technical description of the software in §7.5 is reproduced from a summary written by the assistant that built it. I am including it in that form rather than rewriting it as my own description, because rewriting it would imply an authorship I do not have. It is the most accurate account of that subsystem available, and presenting it accurately matters more to me than presenting it as mine.

In preparing this report, I used an AI assistant to format documents, to reorganize my own written material between sections, and to compute statistics from my exported test data. All observations, decisions, and conclusions recorded in the Learning Log are mine, written at the end of each working day. No AI tool is cited as a source anywhere in this report, because a tool is not a source.

A note on why I worked this way. I set out to build something functional in eleven days with no prior experience in soldering, fabrication, sewing, or embedded software. Writing firmware and a Bluetooth application from scratch was not achievable in that time alongside learning the physical skills, and declining to use an available tool would not have made the project more honest — only smaller. What I did instead was use the assistant where it accelerated me and do the physical engineering myself, which is where the actual learning of this project lives. The clearest evidence for that division is in the failures: every problem I solved by hand — the oxidized tip, the flooded sensor, the battery pads, the harness — I can explain from first principles, because I worked through them, which is not equally true of the software.

12. References

IEEE format. Numbered in order of first appearance in the text.

[1] M. Khalili, S. Lingawi, J. Hutton, J. Christenson, B. Shadgan, B. Grunau, and C. Kuo, "Predicting Heart Rate Estimation Accuracy: Photoplethysmography Sensor Measurements on Eight Anatomical Sites," in 2025 47th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2025. doi: 10.1109/EMBC58623.2025.11253258.

[2] Analog Devices, "ADXL345: 3-Axis, ±2 g/±4 g/±8 g/±16 g Digital Accelerometer," ADXL345 datasheet. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf. [Accessed: Aug. 3, 2026].

[3] Seeed Studio, "Seeed Studio XIAO ESP32C3 Getting Started," Seeed Studio Wiki. [Online]. Available: https://wiki.seeedstudio.com/XIAO_ESP32C3_Getting_Started/. [Accessed: Aug. 3, 2026].

[4] Analog Devices, "MAX30102: High-Sensitivity Pulse Oximeter and Heart-Rate Sensor for Wearable Health," MAX30102 datasheet. [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/max30102.pdf. [Accessed: Aug. 3, 2026].

[5] HackMakeMod, "How to Solder! (Beginner's Guide)," YouTube. [Online]. Available: https://www.youtube.com/watch?v=3jAw41LRBxU. [Accessed: Jul. 21, 2026].

[6] oneTesla, "Soldering Tutorial for Beginners: Five Easy Steps," YouTube. [Online]. Available: https://www.youtube.com/watch?v=Qps9woUGkvI. [Accessed: Jul. 21, 2026].

[7] Last Minute Engineers, "In-Depth: Interfacing MAX30102 Pulse Oximeter and Heart Rate Sensor with Arduino." [Online]. Available: https://lastminuteengineers.com/max30102-pulse-oximeter-heart-rate-sensor-arduino-tutorial/. [Accessed: Jul. 22, 2026].

[8] SparkFun Electronics, "I2C," SparkFun Learn. [Online]. Available: https://learn.sparkfun.com/tutorials/i2c/all. [Accessed: Jul. 23, 2026].

Appendix A — Figure Register

Figures are reproduced in Appendix B. Captions appear beneath each figure.

Appendix B — Photographic Record

The following figures document the build in sequence: components and tools, soldering progression, design and fabrication, schematics, assembly, the finished device, and testing output.

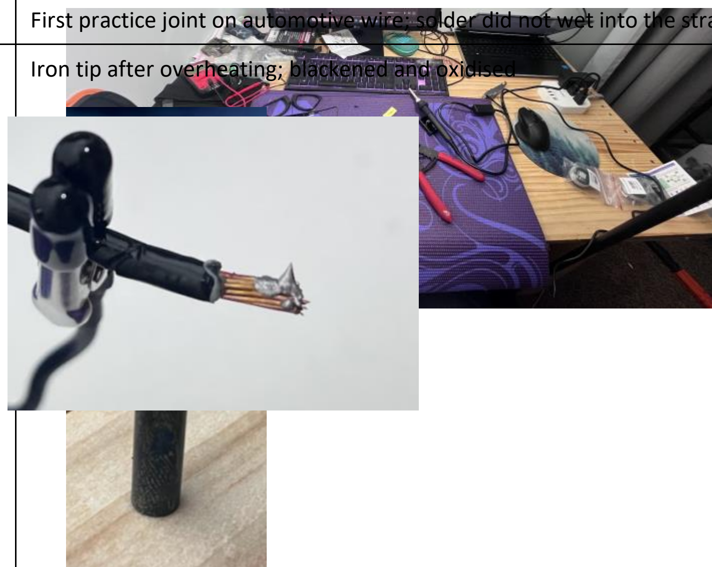
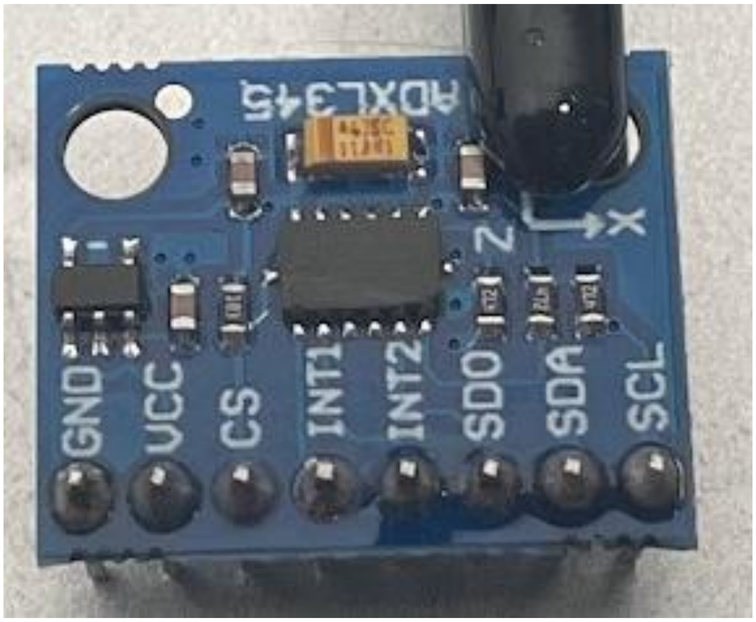
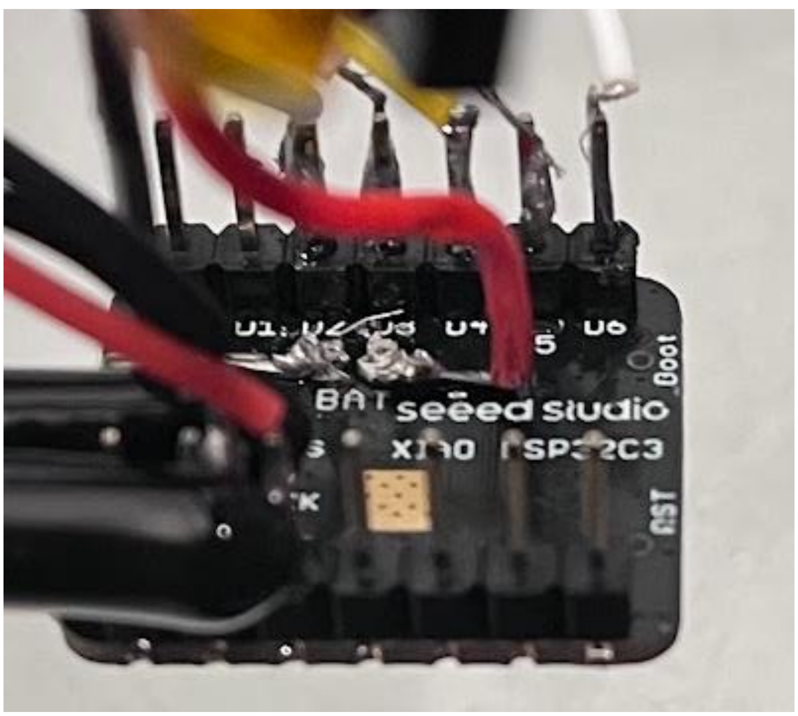
Fig. 1	Workbench setup and soldering tools at the start of the build
Fig.2	First practice joint on automotive wire; solder did not wet into the strands
Fig.3	Iron tip after overheating; blackened and oxidised 
Fig.4	MAX30102 unit 1: bridged pins from over-soldering and damage to the sensor window

	
Fig.5	MAX30102 unit 2: functional joints 
Fig.6	ADXL345 soldered; the cleanest joints of the build

	
Fig.7	Battery terminals soldered to the microcontroller's BAT pads 
Fig.8	Control module drawn in Inkscape

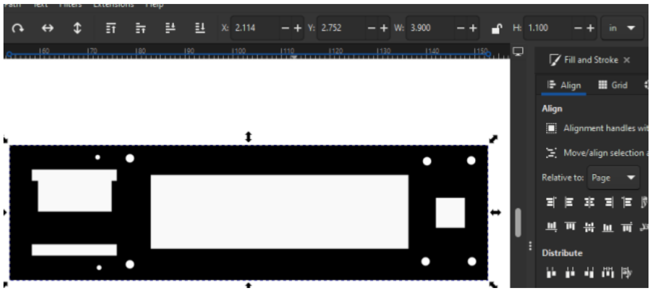
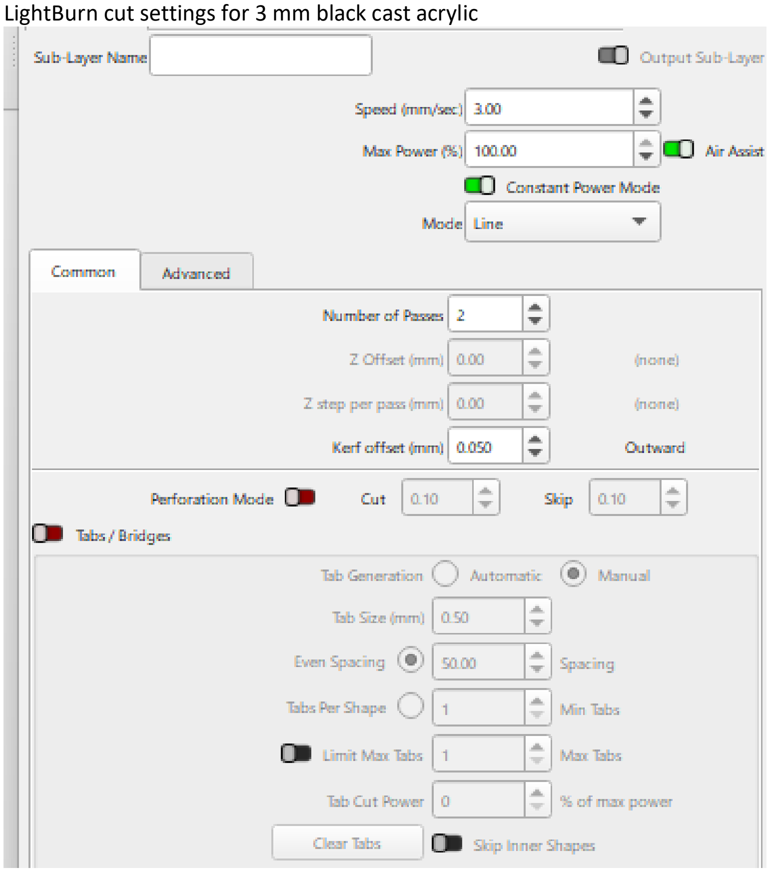
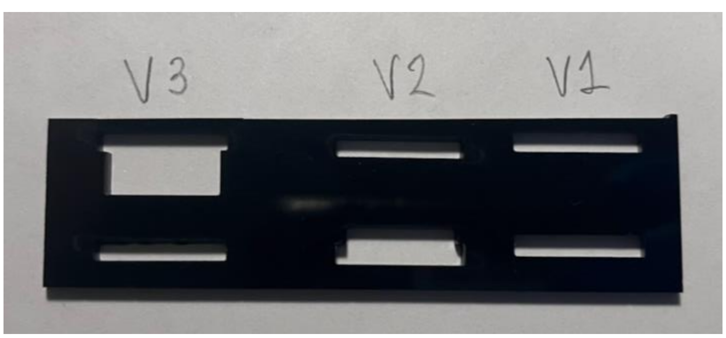
<table>
<tr><td></td><td>
</td></tr>
<tr><td>Fig.9</td><td>LightBurn cut settings for 3 mm black cast acrylic

</td></tr>
<tr><td>Fig.10</td><td>Three friction-fit trials of the XIAO mount cut into a single test board

</td></tr>
</table>

Fig.11 Final control module with microcontroller, battery bay, and button seat

Fig.12 Temple-area sensor mount

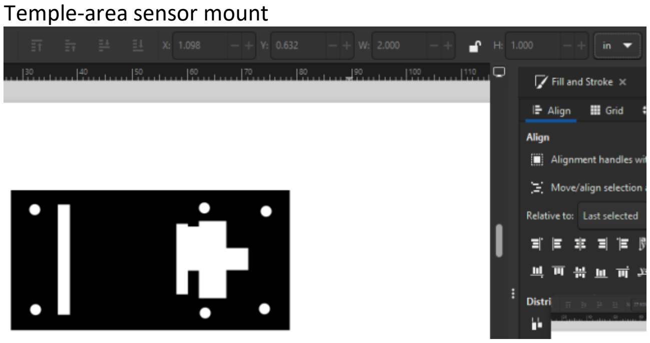


Fig.13 Sewing the fabric housing

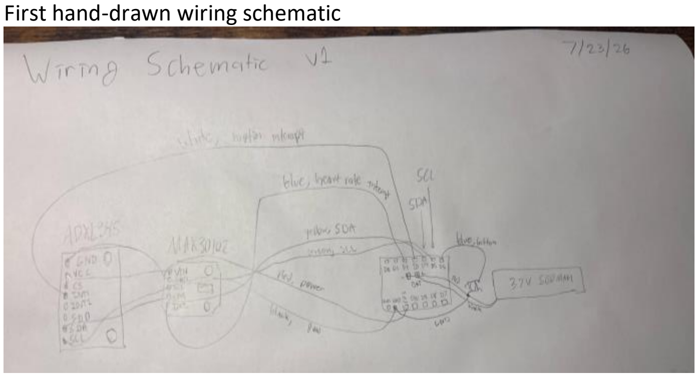
<table>
<tr><td>Fig.14</td><td>Assembled headband with mounting plates sewn in place
</td></tr>
<tr><td>Fig.15</td><td>First hand-drawn wiring schematic

</td></tr>
</table>

Fig.16

Second wiring schematic, drawn roughly to scale

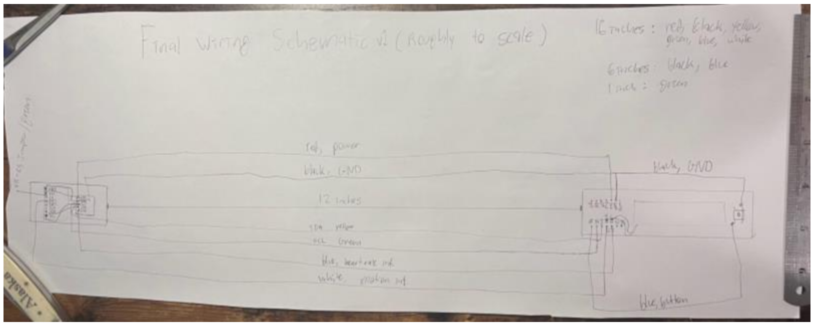


Fig.17

Completed wiring harness with all seven conductors visible

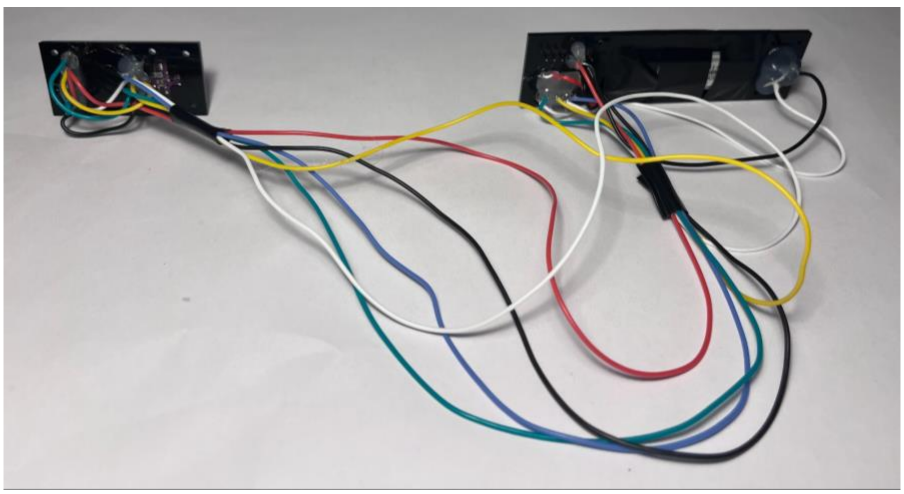


Fig.18

I²C scanner output returning addresses 0x53 and 0x57

```
Scanning...
  device found at 0x53
  device found at 0x57
  2 device(s) total.
Scanning...
  device found at 0x53
  device found at 0x57
  2 device(s) total.
```

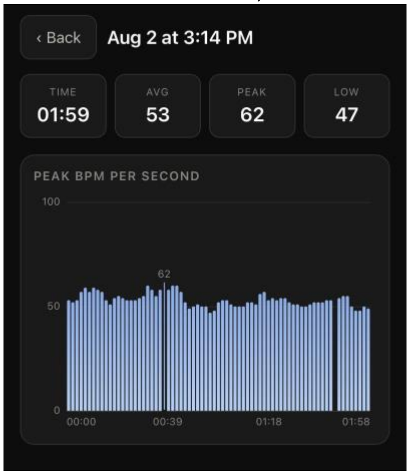
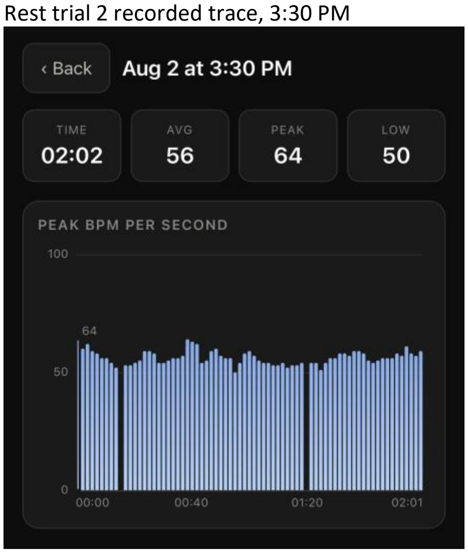
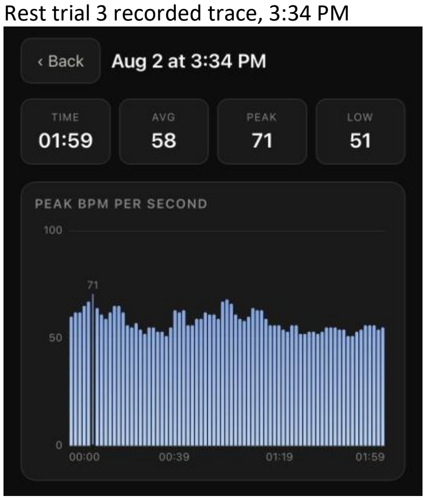
Fig.19	Rest trial 1 recorded trace, 3:14 PM 
Fig.20	Rest trial 2 recorded trace, 3:30 PM 
Fig.21	Rest trial 3 recorded trace, 3:34 PM 

Fig.22

Motion trial 1 recorded trace, 8:54 AM

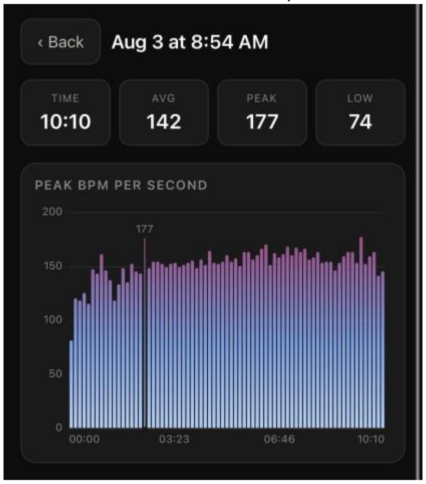


Fig.23

Motion trial 2 recorded trace, 9:19 AM

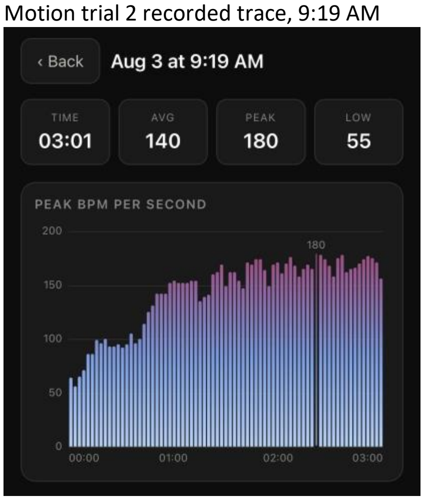


Fig.24

Device worn, front view

Fig.25	Device worn, left side view 
Fig.26	Device worn, right side view 
Fig.27	Device worn, rear view showing the control module