

カレンダーを身に着きたい

-初めてのAI共同ガジェット製作の話-

個人開発集会 2026/09/10
怠けモノ / Namakemono



なぜそんなものを？

来週の水曜って何日？

9月15日って何曜日？



なぜそんなものを？

来週の水曜って何日？

9月15日って何曜日？

→ A. 分かりません



なぜそんなものを？

来週の水曜って何日？

9月15日って何曜日？

→ A. 分かりません

曜日と日付が頭の中で一致せず、
だいたいの予定は曜日で決まっている

→ 日程調整であたふたする二十数年



手元で見れば・・・！

スマホ・・・面倒

腕時計、スマートウォッチ・・・カレンダーではない
パーペチュアルカレンダー・・・携帯が手間



手元で見れば・・・！

スマホ・・・面倒

腕時計、スマートウォッチ・・・カレンダーではない
パーペチュアルカレンダー・・・携帯が手間

→ウェアラブルな
カレンダーを作ろう！



要件定義

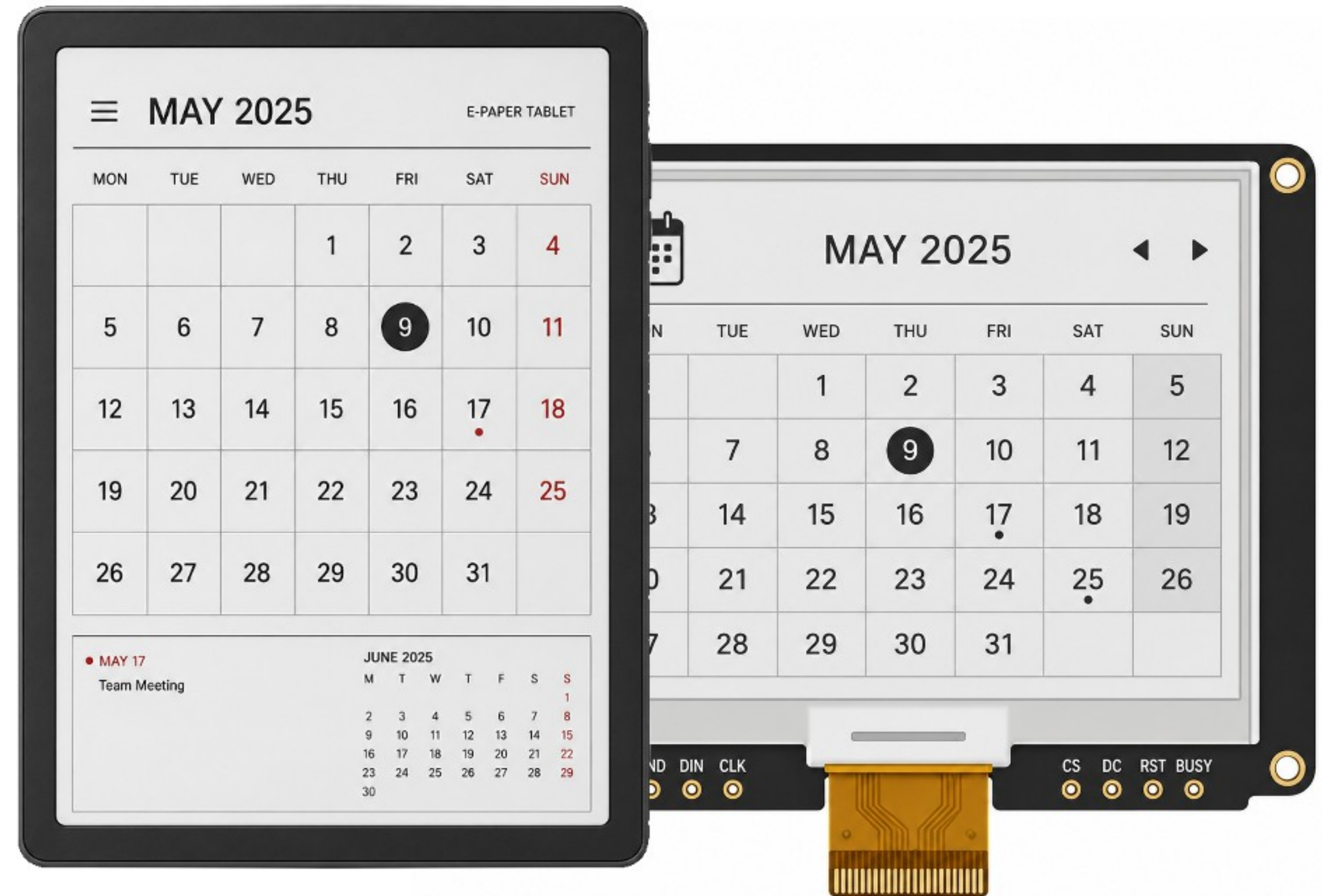
- ・ 腕に装着できる
- ・ 今週を中央に前後5週間
- ・ ひと目で読める
- ・ 見るための操作がいらぬい
- ・ 充電、メンテナンスを
極力減らす

要件定義

- ・ 腕に装着できる
- ・ 今週を中央に前後5週間
- ・ ひと目で読める
- ・ 見るための操作がいらない
- ・ 充電、メンテナンスを
極力減らす

→ E-paperを選定

- ・ 常時表示なので操作不要
 - ・ 表示保持に電力不要
- 腕側のバッテリーをなくせる



選定部品

ディスプレイ

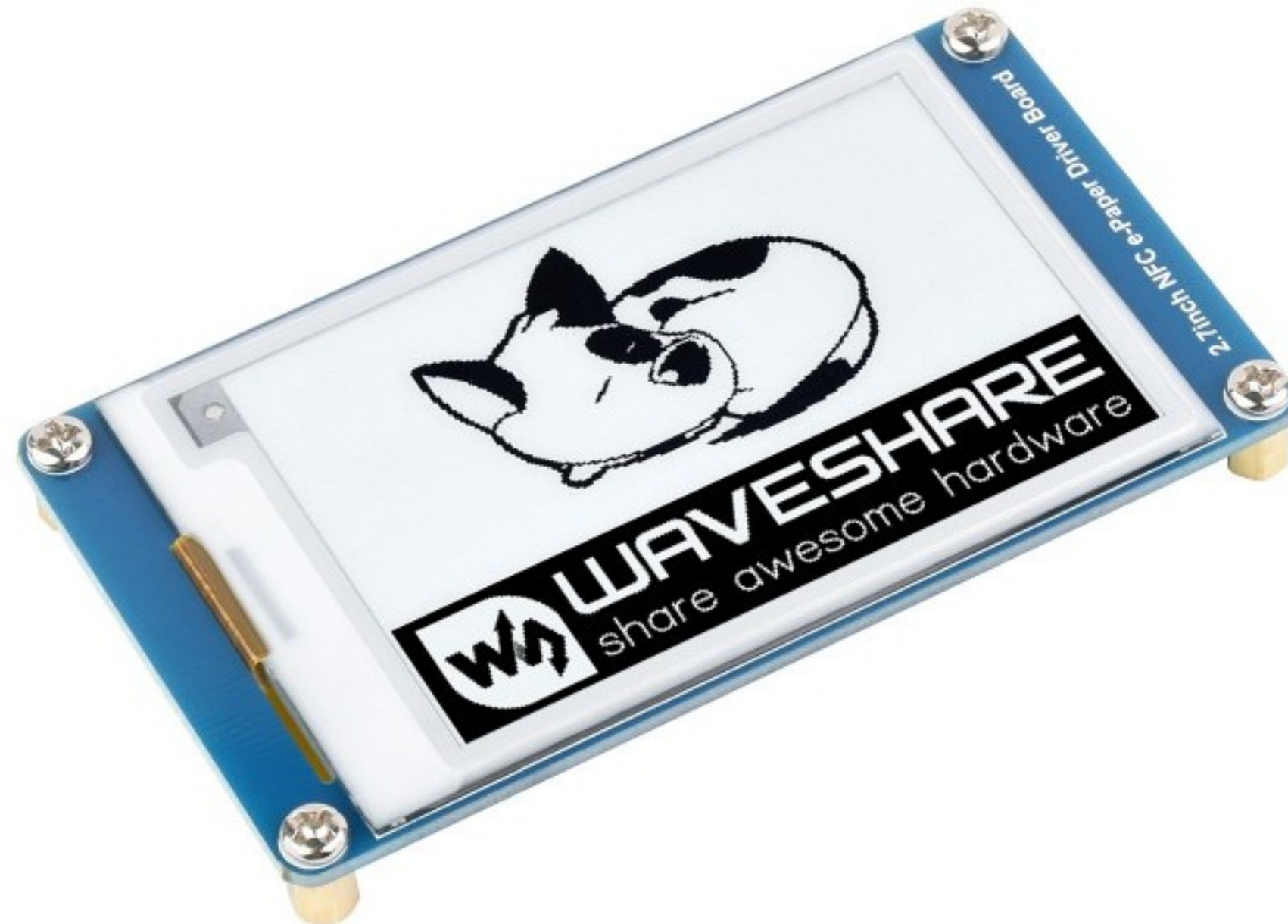
- ・ Waveshare 2.7インチ e-Paper(一作目)
- ・ Waveshare 1.54インチ e-Paper(二作目)

装着部

- ・ NATOバンド
- ・ PETG(白)

選定理由

- ・ 両方ともNFC電力のみで書き換え可能
- ・ 2.7インチは外装無し



2.7インチの失敗

ソフトウェアは完成！

2.7インチ e-Paperを装着した結果・・・

2.7インチの失敗

ソフトウェアは完成！

2.7インチ e-Paperを装着した結果・・・

でかくね・・・？

腕には載るが、
バンドを通す余白がない



1.54インチの再選定

H46.9 W41.4 D6.6(mm)

3色表示(赤、白、黒)

ケース付きで扱いやすい

**→こっちの方が
カレンダー向きでは？**



アプリの再設計

画面が小さくなったのでUIも作り直す
機能を絞りつつ直近の
予定を見れるように改良

- ・ 休日を赤文字に設定
 - ・ カレンダーやGoogleタスクを連携可能
 - ・ GithubページにアップロードしたHTMLを参照
 - ・ バージョンを確認して更新も可
- UI等の改良がアプリ更新を挟まずに可能

NFCカレンダー 1.54

	S	M	T	W	T	F	S
8	16	17	18	19	20	21	22
	23	24	25	26	27	28	29
9	30	31	1	2	3	4	5
	6	7	8	9	10	11	12
	13	14	15	16	17	18	19

11/1(日)
・ 10:00 案丸点検

予定モード

休日モード

日付を選ぶたびに、予定枠の有無を切り替えます。

基準日

2026/08/30

← 前週

今日

次週 →

カレンダー予定 4件を反映

「Googleタスクを連携・更新」を押してください

Googleタスクを連携・更新

NFCで転送

PNGを保存

ログをコピー

クリア

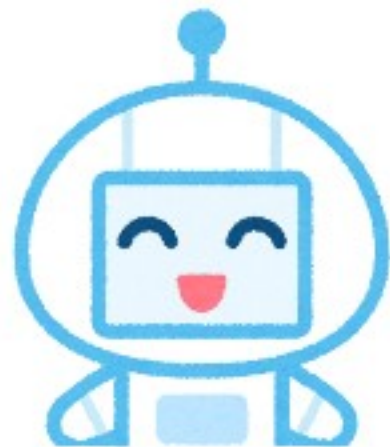
HTML v2026.08.10.5

[1:40:44] 1.54インチ カレンダーHTMLを読み込みました

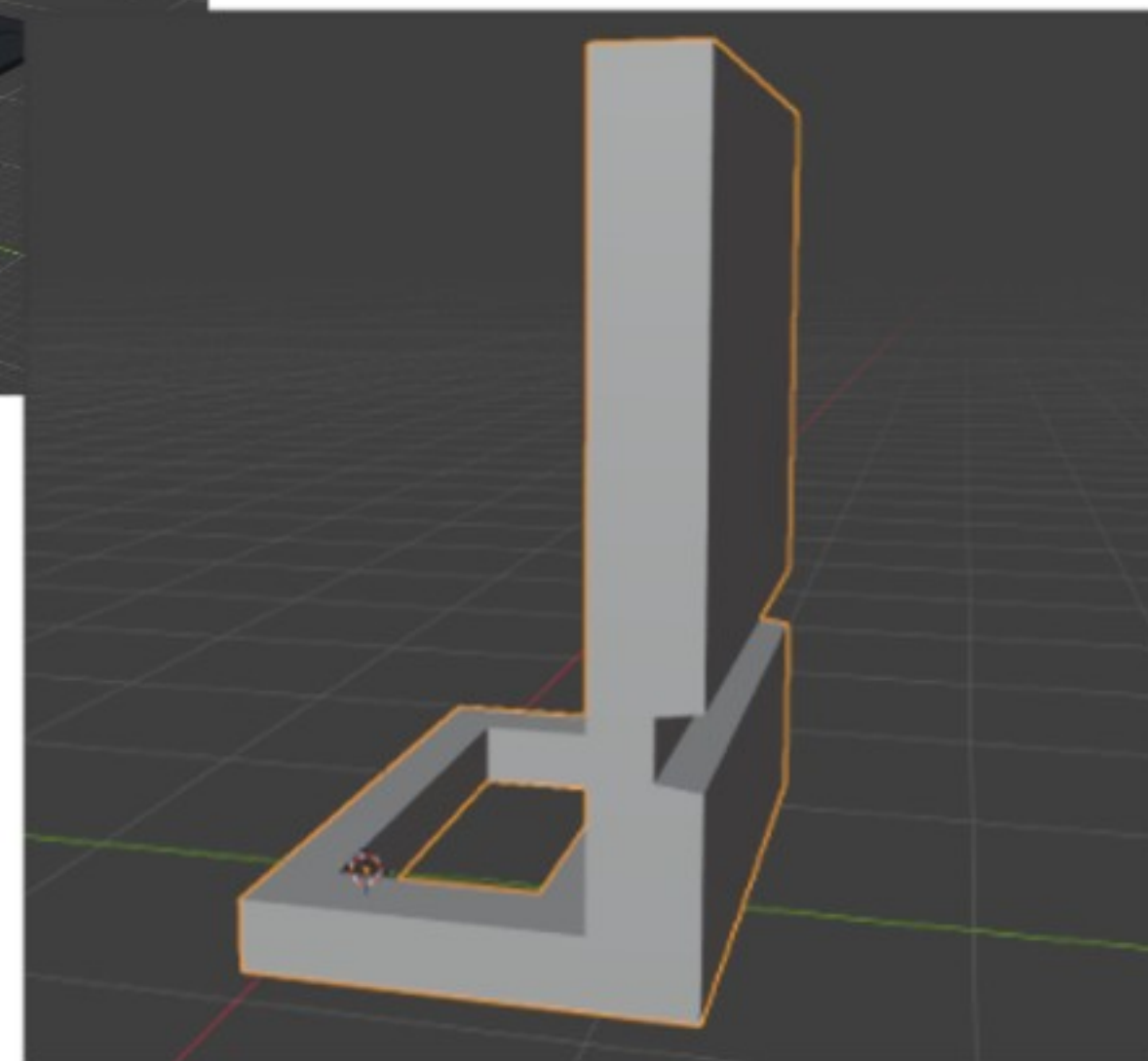
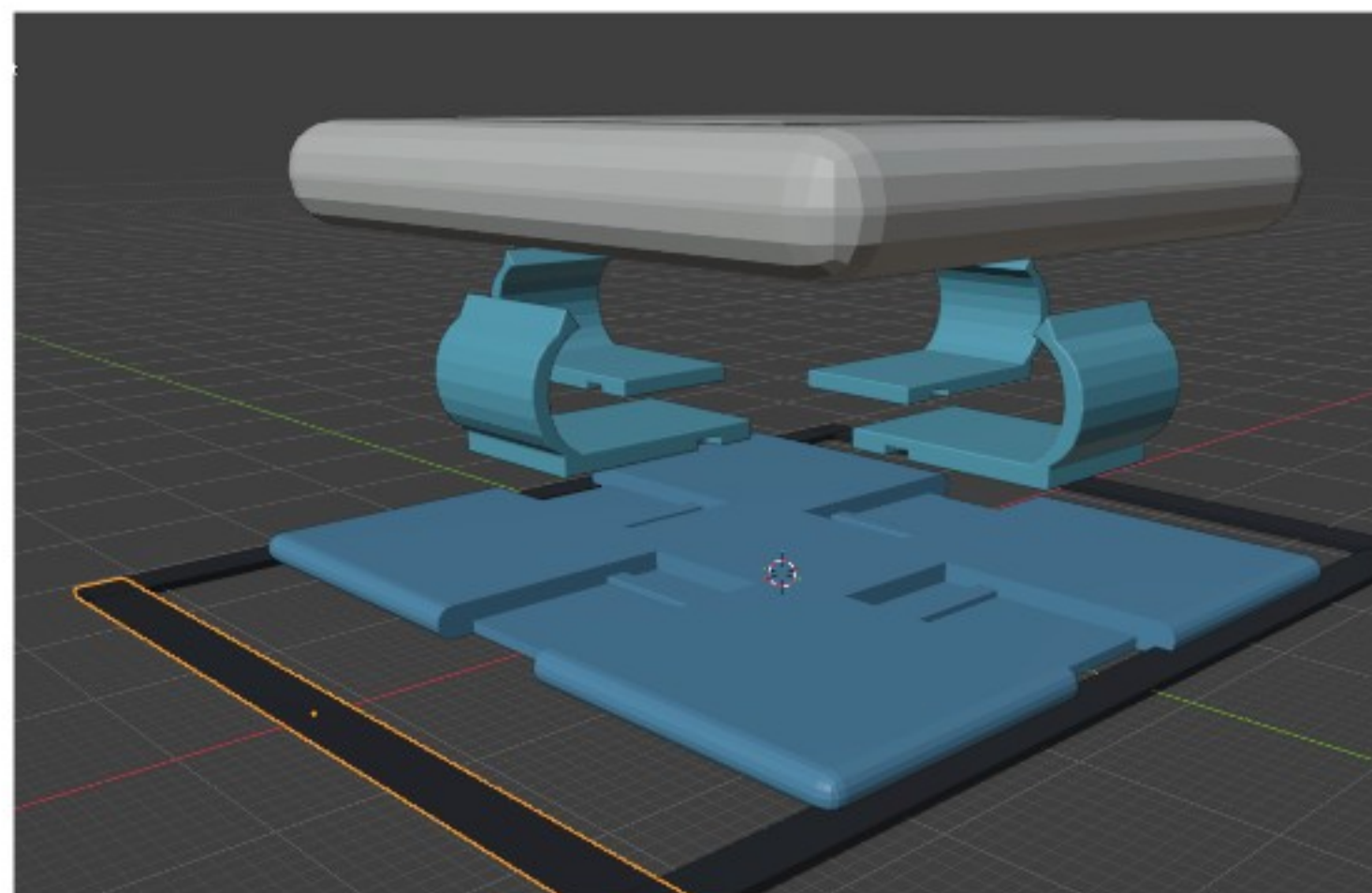
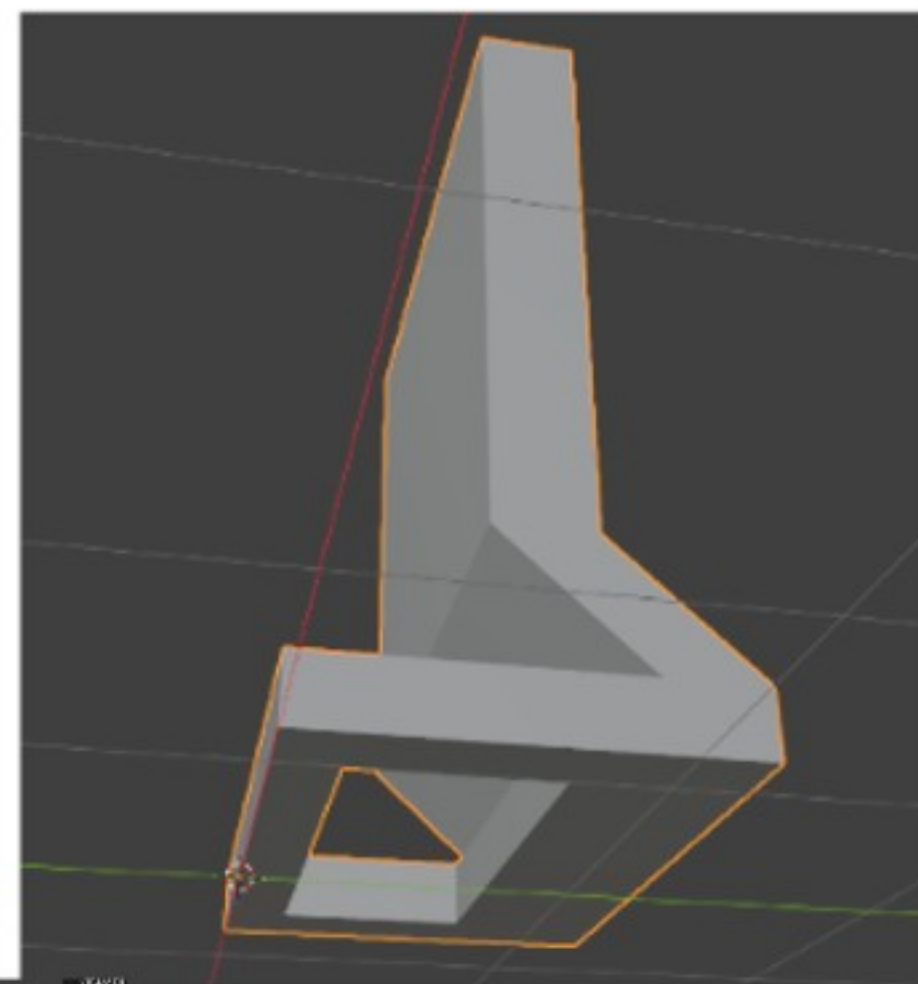
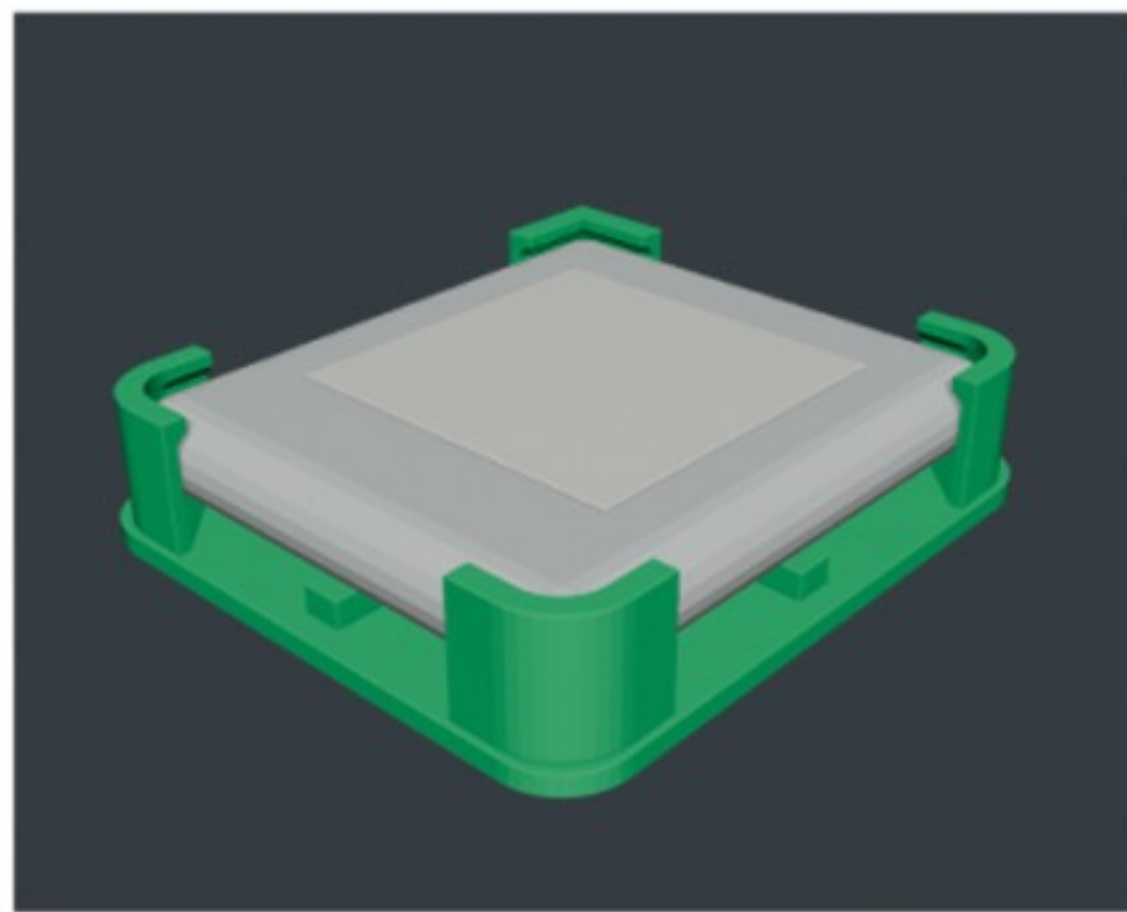
[1:40:45] カレンダー予定件数=4

腕に付ける

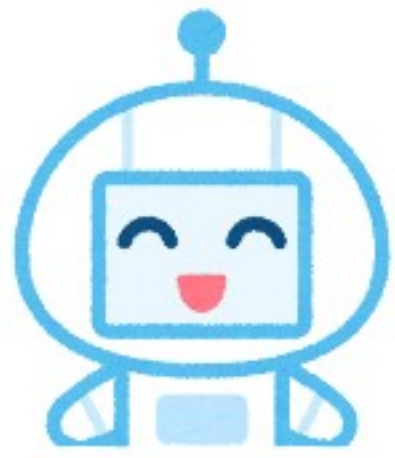
腕に付ける 茨の道でした



溝にして→しない
スペース開けて→あけない
外向きに傾けて→逆にしちゃう
→2週間くらいかかった

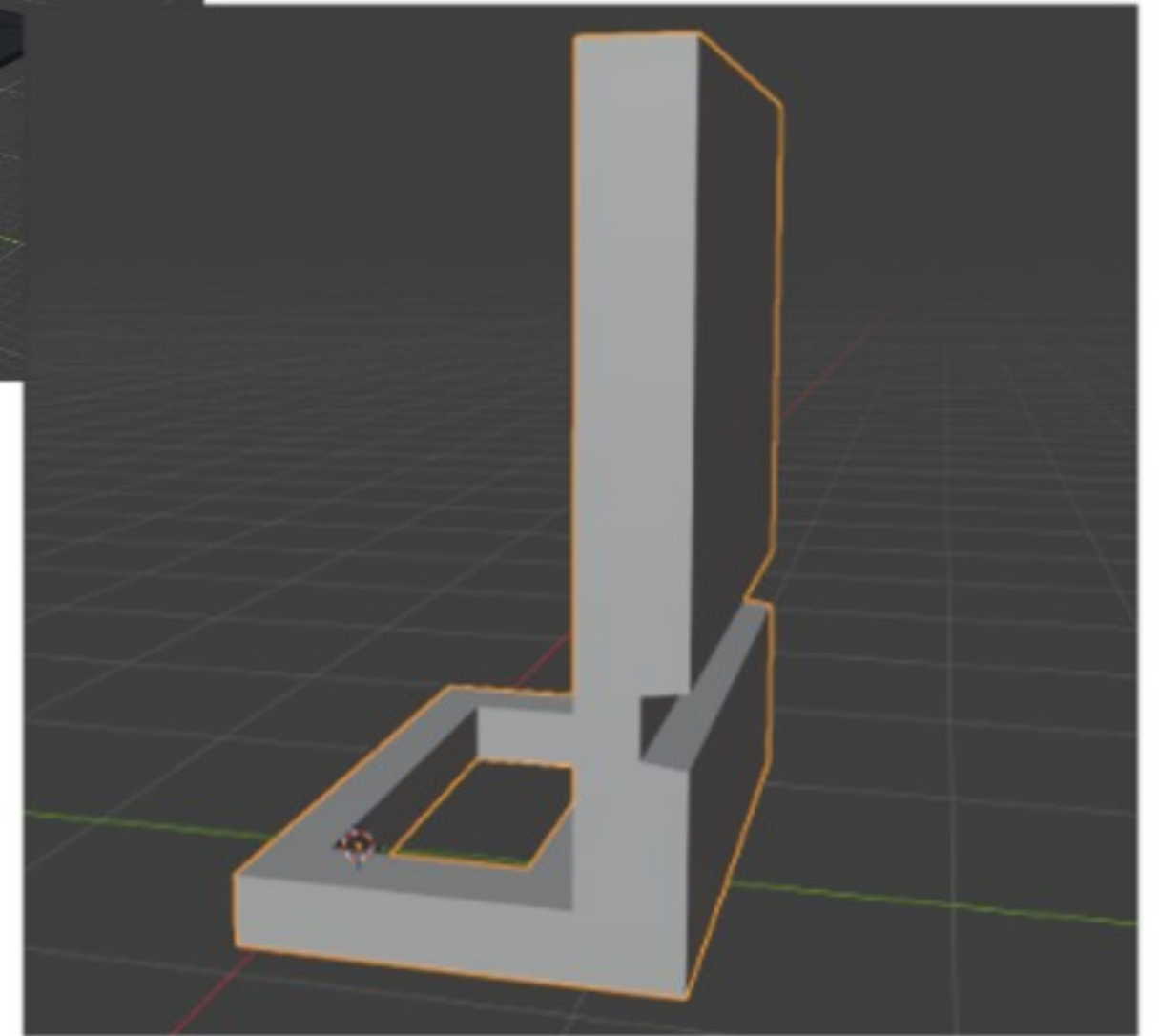
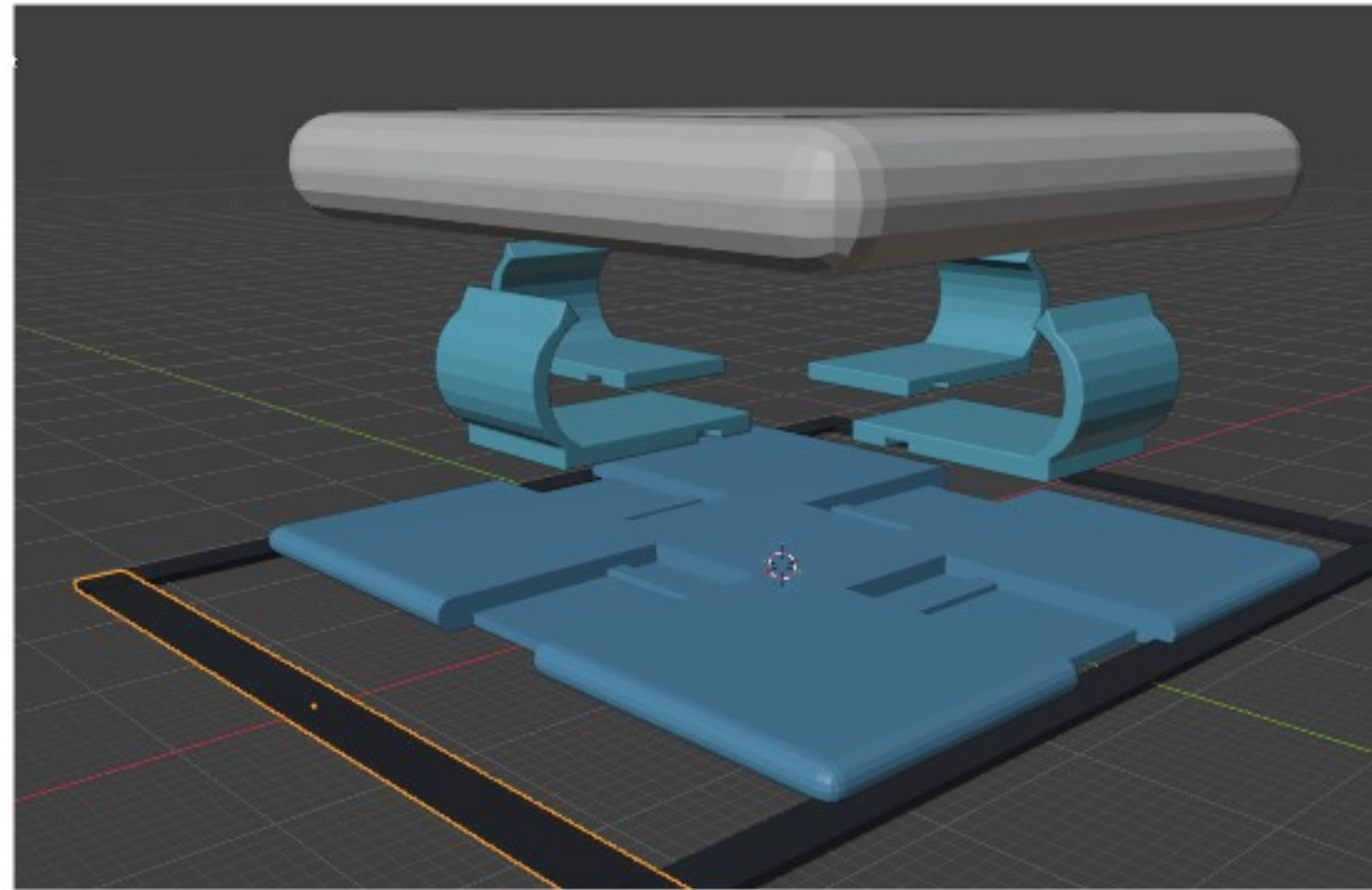
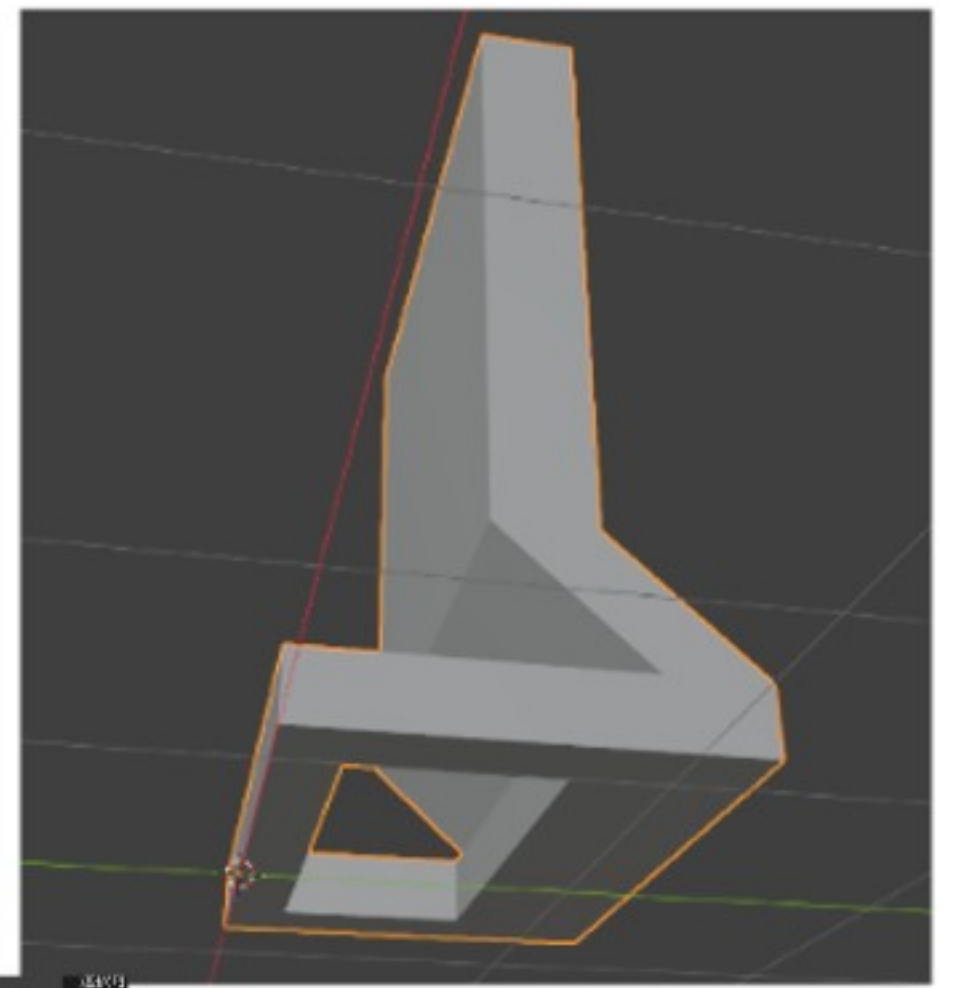
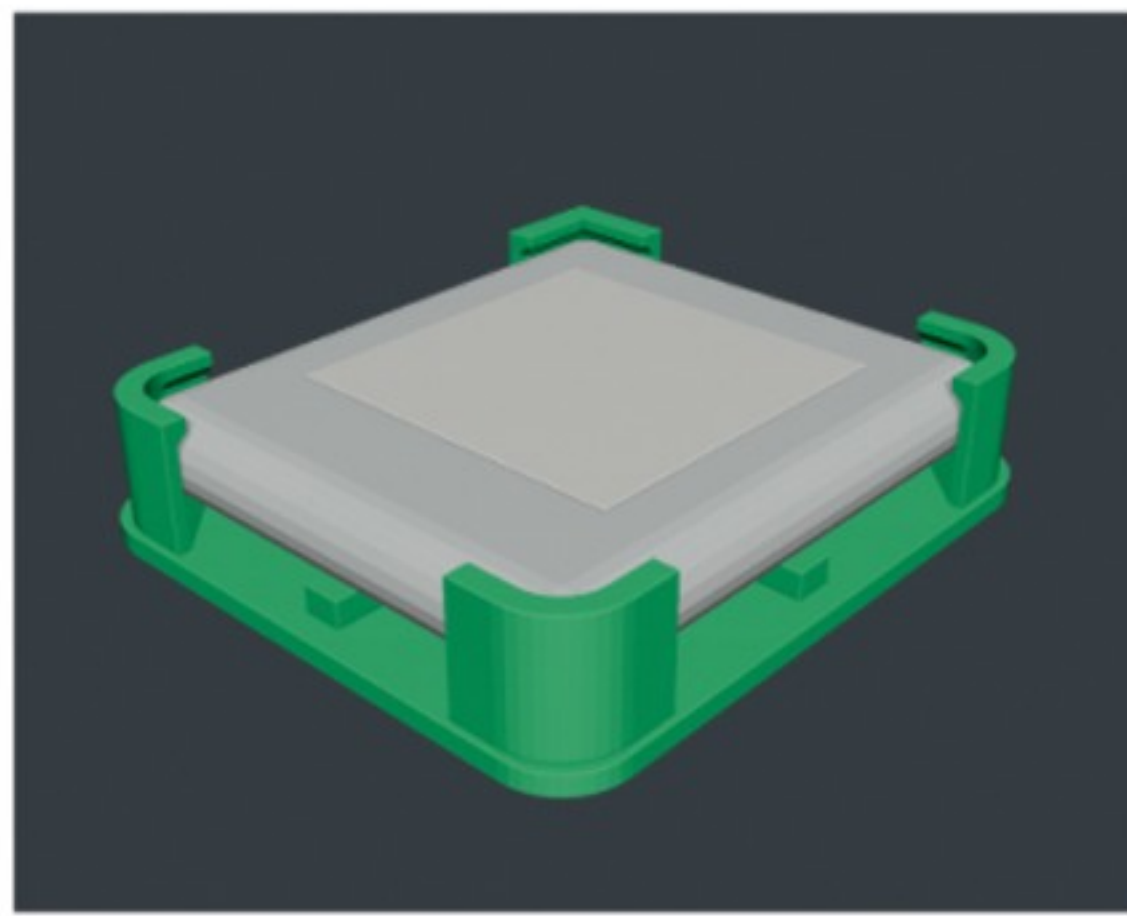


腕に付ける 茨の道でした



溝にして→しない
スペース開けて→あけない
外向きに傾けて→逆にしちゃう
→2週間くらいかかった

確実に自分で作って
プリントしながらテストした方が速かった



AIが得意だったもの

ソフト

- ・ 公式SDKを解析
- ・ 組み合わせ総テスト環境を構築できる
- ・ 必要な情報をログに残して原因説明
- ・ SDKが対応してない事まで突き止められる
- ・ その上で対応する処理を実装できる

AIが得意だったもの

ソフト

- ・ 公式SDKを解析
- ・ 組み合わせ総テスト環境を構築できる
- ・ 必要な情報をログに残して原因説明
- ・ SDKが対応してない事まで突き止められる
- ・ その上で対応する処理を実装できる

一方ハードは・・・

- ・ 「上に付けて!」→「分かりました」
→下 に 生 や す
- ・ 「逆にして!」→「修正しました」
→別 の 部 位 を 逆 に す る
- ・ 「これ参考にして!」→「参考にしました」
→ど こ を ？ ？ ？

コードは強い。立体は難しい

コード

正解条件を文書や数値で伝えやすく、
AI側も判断しやすい。

立体

「写真向かって右」「下側」「幅」が視覚情報依存

→現物を触るものは、人間が直接直した方が速いこともある

→文章・数値で正解を定義できる作業はAIに任せやすい
身体寸法・しなり・装着感は現物確認が必要だった

開発状況

・ 現在地

- ・ 第一段階のアプリ開発完了
- ・ NATOバンド交換機構も一旦完成

→ カレンダーを身に着けられるところまで来た

・ 今後の予定

- ・ Googleタスクとの連携確認
- ・ TPUによる防水ケースの製作
- ・ フレキシブルE-paperにNFCアンテナ実装の検討

→ もっと「今日何日？」に困らない腕へ