

Raspberry Pi Pico を用いた無線マウスの製作②

1. はじめに

「ヒューマンインターフェース」という言葉は人と機械の関りを表すだけでなく、機械を通じて人ととの関りを考えるという理念が込められている。Apple ではガイドライン(図 1)を設け、それに沿って iOS のデザインを決めている。

図 1 Apple Developer(2025/4/19)

無線マウスの製作を通して、入力機能の設計について学び、「人と人とをつなぐデバイスのこれから」を考えてもらいたい。

2. 目的

Raspberry Pi Pico W と Arduino IDE を利用してスイッチング入力やアナログ入力を制御する。

3. 基礎知識

① Human Interface

1980 年代後半から 90 年代にかけて、機械は性能の良さだけでなく「人に分かりやすい」「人に優しい」ことの重要性が認識されて、ヒューマンインターフェース技術が生まれた。ものを製作し技術を提供する立場からではなく、技術の受け手側の立場からの研究・開発を行う。技術的には理学工学が基礎となるが、人の原理を「生理」「認知」「心理」「文化」「社会」などの視点で取り入れ、「感性」「魅力」を考えたデザインを行うことが求められている。

図 2 ヒューマンインターフェース技術の推移

② Raspberry Pi Pico W

ARM マイクロコントローラチップ RP2040 を搭載したマイコンボード Raspberry Pi Pico に infineon CYW43439 が追加され、Wi-Fi(IEEE 802.11b/g/n)と Bluetooth(v5.2)が利用可能になった。GPIO(General-purpose input/output)、ADC(Analog-Digital Converter)、外部電源入力や定電圧出力ポートなどを持つ。

表 1 Raspberry Pi Pico W の主なスペック

CPU	Cortex-M0+ 133MHz × 2
RAM	264kB on-chip SRAM
Flash Memory	2MB on-board Quad SPI
Power supply	1.8 – 5.5 V
USB	Micro-USB
Size	51mm × 21mm × 3.9mm / 3g

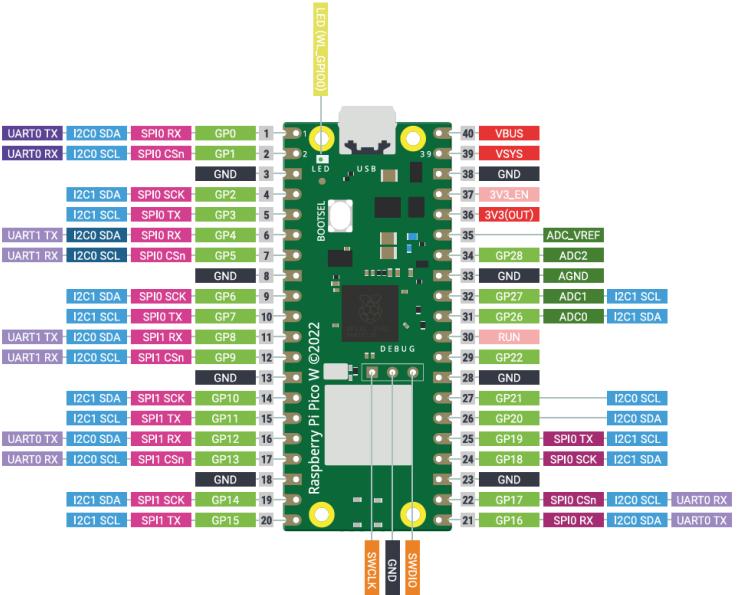


図 3 The pinout of the Pico W Rev3 board

③ 工事設計認証(電波法第38条の24)

工事設計認証とは、特定無線設備が技術基準に適合しているかどうかの判定について、その設計図(工事設計)及び製造等の取扱いの段階における品質管理方法(確認の方法)を対象とし、登録証明機関が行う認証制度。

無線設備そのものではなく、工事設計を対象としており、実際の無線設備は認証後に製造される点が、技術基準適合証明と異なる。技適マークは、工事設計認証を受けた者(認証取扱業者)が付する。

技適未取得機器を使用した場合、電波法第110条に基づき(　　)年以下の懲役または(　　)万円以下の罰金が科される可能性がある。



4. 実習内容

① 開発環境の構築

(a) PCへのログイン

(b) デバイスピライバーセットアップの選択

→すべて「いいえ」 → 同意

(c) Arduino起動

mdns-discovery → 許可

dpinst-amd64.exe → 許可

※管理者権限が必要になりますので

必要に応じて教員が入力します

項目	仕様・型番
CPU	
RAM	
OS	
Arduino IDE	
Boad	
Library	

(d) ボードの追加

File → Preferences →

Additional boards manager URLs

→ Raspberry Pi RP2040 Boards

Additional Boards Manager URLs

Enter additional URLs, one for each row

Raspberry Pi RP2040 Boards: https://github.com/earlephilhower/arduino-pico/releases/download/global/package_rp2040_index.json

All major RP2040-based boards

(e) ライブラリの追加

Sketch → Include Library →

Manage Libraries →

「ESP32 BLE Arduino」で検索

ESP32 BLE Arduino by Neil Kolban
<kolban1@kolban.com>

1.0.1 installed

BLE functions for ESP32 This library provides an implementation Bluetooth Low Energy support for the ESP32 using the Arduino platform.

More info

1.0.1 REMOVE

(f) ボードの選択

「Raspberry Pi Pico W」選択

File Edit Sketch Tools Help

Raspberry Pi Pico W

(g) ネットワークの設定

Tools → IP/Bluetooth Stack
→ IPv4 + Bluetooth

(h) ターゲットボードへの書き込み

- ・書き込みモード
- BOOTSW : ON で電源投入
- RSTSW : ON → BOOTSW : ON → RSTSW : OFF → BOOTSW : OFF

- ・書き込み
LED 点滅を確認

② 演習 1 (GPIO の設定)

回路図から GPIO を割り当て、指定の領域にプログラムしよう。

変数型	変数名	GPIO	役割
const int	RIGHT	20	右クリック
const int	LEFT	21	左クリック
const int	Xaxis	26	X 座標増減
const int	Yaxis	27	Y 座標増減

③ 演習 2 (座標の設定)

X 座標の最大値と Y 座標の最大値を、指定の領域にプログラムしよう。



変数型	変数名	値	役割
const int	X_MAX		X 座標最大
const int	Y_MAX		Y 座標最大

④ 演習 3 (変数の初期設定)

ジョイスティックの中立位置と移動距離の初期座標を、指定の領域にプログラムしよう。

変数型	変数名	値	役割
int	x_0	0	X 座標中立
int	y_0	0	Y 座標中立
int	x_val	0	X 座標距離
int	y_val	0	Y 座標距離

⑤ 演習 4 (Bluetooth 接続)

(a) マウスの BLE エミュレート開始

```
MouseBLE.begin(" mouse**** ");
```

****には学年組出席番号 4 行を入力

(b) キーボードのエミュレート開始

```
Keyboard.begin();
```

マウス・キーボードのエミュレート開始命令を指定の領域にプログラムしよう。

⑥ 演習 5 (GPIO と ADC の初期設定)

(a) Analog 入力分解能

```
analogReadResolution(分解能);
```

(b) GPIO 入出力設定

```
pinMode(入出力設定);
```

変数名	入出力
Xaxis	INPUT
Yaxis	INPUT
RIGHT	INPUT_PULLUP
LEFT	INPUT_PULLUP

Analog 入力分解能(12bit)、GPIO 入出力設定を指定の領域にプログラムしよう。

課題1 ジョイスティックを調整しよう

`analogRead(Xaxis);`と`analogRead(Yaxis);`でジョイスティックの位置情報をfor文でDIV回読み取り、`x_sum`にX座標の総和、`y_sum`にY座標の総和を保存する。

繰り返しの度にLEDを点滅させることによりキャリブレーションの実行が視認できるようにしよう。

```
digitalWrite(LED_BUILTIN,HIGH);
delay(50);
digitalWrite(LED_BUILTIN,LOW);
delay(50);
```

図4 点滅プログラム例



課題2 ジョイスティックの状態を見よう

動作確認用のwhile文を消そう。

シリアルモニタに

「x_0」「y_0」「Xaxis」「Yaxis」「X」「Y」の値を表示させよう。

・文字列の出力 `Serial.print("文字列");`

・変数値の出力 `Serial.print(変数名);`

```
x_0:2134|y_0:2142|Xaxis:2131|Yaxis:2136|X:0|Y:0
x_0:2134|y_0:2142|Xaxis:2133|Yaxis:2136|X:0|Y:0
x_0:2134|y_0:2142|Xaxis:2132|Yaxis:2136|X:0|Y:0
x_0:2134|y_0:2142|Xaxis:2128|Yaxis:2132|X:0|Y:0
x_0:2134|y_0:2142|Xaxis:2131|Yaxis:2131|X:0|Y:0
x_0:2134|y_0:2142|Xaxis:2128|Yaxis:2139|X:0|Y:0
x_0:2134|y_0:2142|Xaxis:2129|Yaxis:2134|X:0|Y:0
```

図5 出力例



課題3 カーソルの有効範囲を設定しよう

マウスカーソルが画面の端にかかると、座標が更新されずにその場から一切動けなくなってしまう。この現象を回避するため以下のような考え方がある。

X座標 `x_val` が 0 未満ならば

→ `x_val` に 0 を代入

X座標 `x_val` が 11000 を超えたなら

→ `x_val` に 11000 を代入

Y座標 `y_val` が 0 未満ならば

→ `y_val` に 0 を代入

Y座標 `y_val` が 11000 を超えたなら

→ `y_val` に 11000 を代入



課題4 オプションボタンで文字列表示

カーソルは動くようになったけど、何だか物足りない…そんなあなたにオプションボタンを用意しました。

このボタンを押して、秘密の文字列が瞬時に打ち込まれるようにしてみよう。

```
Keyboard.print("秘密の文字列");
```

打ち込まれた文字列はENTERキーで決定

```
Keyboard.write(KEY_RETURN);
```

※ この課題は指定された領域外にもプログラムを追加する必要があります。

