

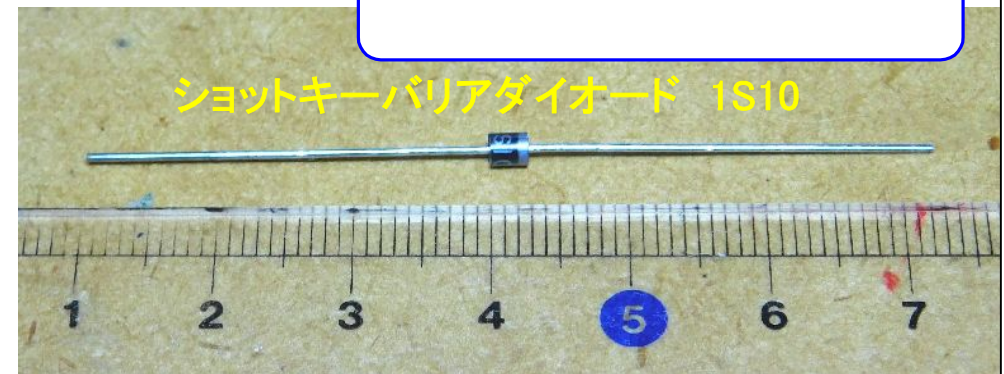
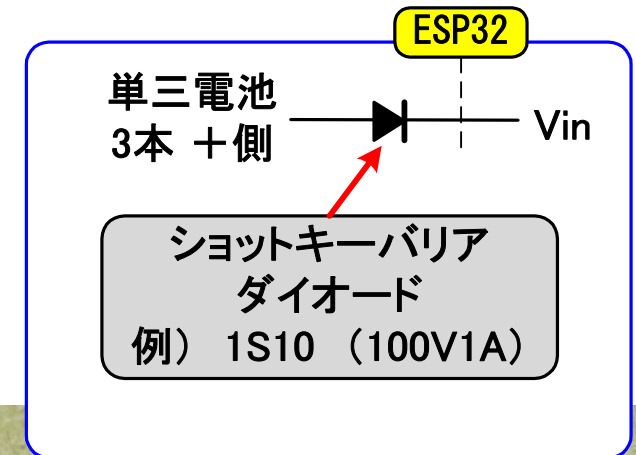
追加、簡易ロガーの電池駆動

USBケーブルから、5Vの電源を供給しつつ、もう一つ、5V別電源を接続するために 別途 5V 電源出力に ショットキーバリアダイオード を接続して 2つの 電源を 供給し、その場合は USB側の 5Vが、ESP32 CPUに 供給され

USBケーブルを引き抜けば 別途 5V電源から ショットキーバリアダイオード経由で ESP32 CPUに 電源が供給される回路を 以前 157 の 動画にて説明しました。

で、別途 5V電源を 単三乾電池 直列3本に置き換えると 通常 単三乾電池一本は 1.5V出力と言われますが、新品時は 1.6Vぐらい出ますので 3本で、4.8Vぐらいになります。

4.5Vで 考えても ショットキーバリアダイオードで、電圧が 0.3Vドロップして 4.2Vになります。ESP32マイコンは 3.3Vで動作しますが



3.3Vを出力するために LDOの 三端子電源 IC が 接続されています。電圧ドロップは 0.6V程度なので3.9V以上あれば ESP32マイコンは動作する事になります。ショットキーバリアダイオード出力が 4.2Vであれば 一応動きます。

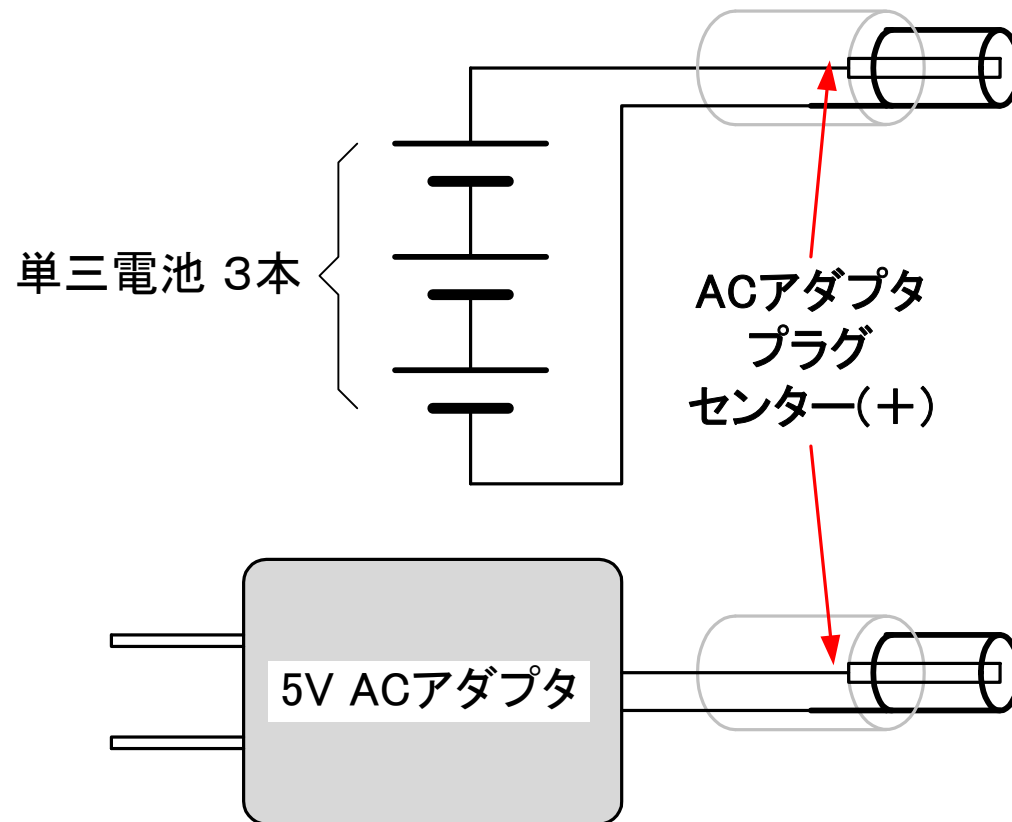
尚、ESP32基板モジュールの 3.3V端子に、3.3Vの モバイル用リチウムバッテリーの出力を接続するのは、やめて下さい。ESP32 CPUには 支障が無いと思いますが、**LDOの三端子電源ICの出力端子に、リチウムバッテリーの電流が 逆流する事になり 三端子電源ICが 壊れる事になる**と思います。どうしてもやりたいのであれば、三端子電源ICの 3.3V出力端子を 切り離して下さい。その場合、その後は 基板上の三端子電源IC出力が ESP32 CPUに 3.3V電源を 供給出来ませんので、**使用する時は、常時 3.3Vを外部から供給して下さい。**尚、この改造は私は **責任を負いませんので 自己責任で行って下さい。** というか、**基本電子工作は 自己責任で行う物です。**

ESP32基板モジュールにて 3.3V端子に 外部から 3.3Vを供給する事に関しては 以上です。

話が 横道にそれましたが ESP32基板モジュールの 5V端子部分で ダイオードで 逆流防止も兼ねて 5Vより少し低い電圧で、電圧供給する方法で、話を進めます。単三電池3本で、ESP32マイコンの単独動作を行わせる事もあれば、電源の取れる場所で、5Vの ACアダプタを 長期使用的に使う場合もあるかもしれません。

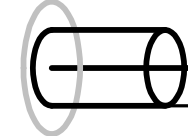
そういう事を考えると ESP32モジュール基板の5V端子側に ACアダプタを接続するジャックを 付ける事にして ジャックと 5V端子の間に ショットキーバリアダイオードを 入れるのが いいと思います。単三電池3本側の出力には ACアダプタで使うプラグを 付ける必要があります。そうすれば、用途に応じて 電池を接続するか、ACアダプタを接続するかの選択が 出来ます。次ページで、接続の回路図を示します。

接続確認用 回路図



ACアダプタ
ジャック

1S10



Vin (5V)

Gnd

ESP32

一時的に
この部分に
電流測定用の
低い抵抗値の
抵抗を入れる

ACアダプタジャックに 単三電池3本
か 5V ACアダプタの どちらかを接続
する。

ESP32側に 押しボタン入力を1回路追加

パソコンと切り離れた状態で、電池駆動で運転モードに なっていると、移動時の振動とかで、余分な振動を データとして記録するのでデータを収録したいところに持って行くまでは停止モードで 収録したい現場に到着しセンサを設置したら 運転モードにする。

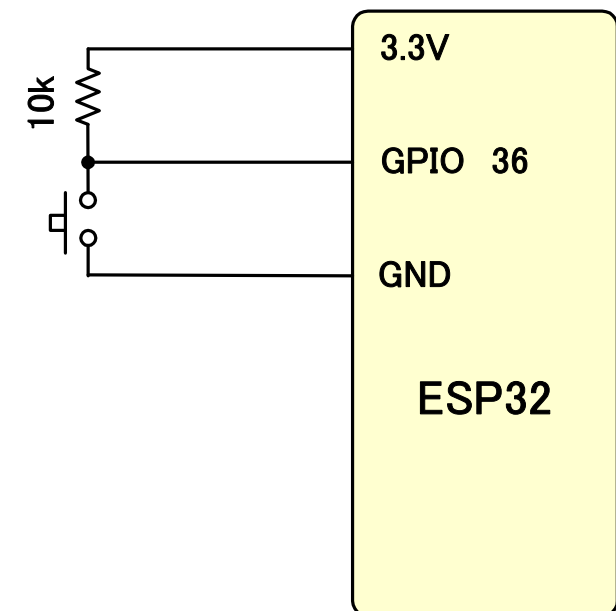
逆に目的の振動を収録し 帰宅する際には、また 停止モードにして持ち帰る。

あるいは、帰宅する場合は、もう すぐに収録する事は 無いでしょうから、電池の電源を 抜いて ESP32を 完全に電源OFF状態にして持ち帰ってもいいでしょう。

何らかのトラブル発生に対応するため、可能であればノートPCも 現場と一緒に持って行く方が 安心感は あります。

で、運転モードと 停止モードを切り替える押しボタンスイッチですが 入力専用のGPIO36 端子に 接続する予定です。
大雑把な、回路図を示します。

運転、停止 モード切替えスイッチの 追加



運転、停止ボタンの 読み取り処理

モーメンタリの押しボタンスイッチを使用する予定です。停止時に 1回押すと 運転モードに移行、再度押すと 停止モードに移行という処理に します。

この場合は、スイッチのチャタリングは しっかり取っておく必要があります。

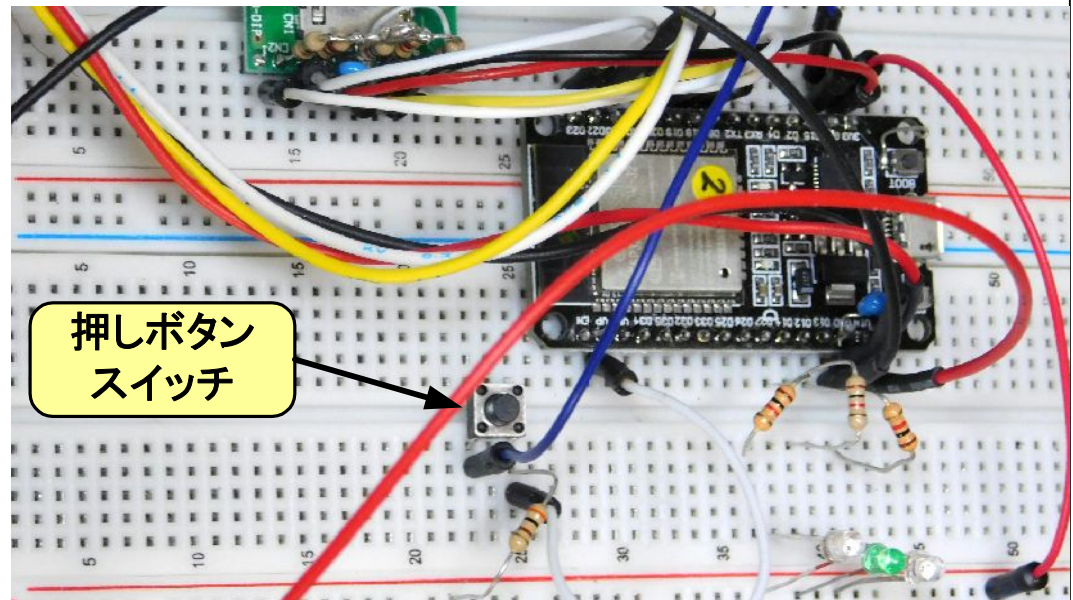
100Hzタイマー割り込みで、GPIO36の状態を読み込み、スタティックな byte 変数の最下位 bit に、スイッチ接点が HIGHなら 1、LOWなら 0 を上位 bit が 変わらないようにして 入れます。

そして 1 bit 左シフトします。そして、3回連続して読み取り 下位 4 bit が 1000 ならスイッチが 押されたと判断します。

という事で、まずは 金物を用意します。

その後、運転、停止 押しボタンSWの 関数を ESP32プログラムに 追加します。

動画は、完成後の 実験の所から始めます。



用意した金物

電池以外は、昔、秋月電子から購入した在庫品です。

上左から

- ① ACアダプタプラグ
- ② ACアダプタジャック
- ③ 1S10 ショットキーバリアダイオード
- ④ 5.1 Ω / 1W 抵抗
- ⑤ 単三電池3本用電池ボックス
スイッチ付き

ここの画像には含まれていませんが
⑥ 押しボタンスイッチ
(タクトスイッチ)と PullUp抵抗が必要です。

