

## **Selbstbau (DIY)-Fernbedienung für Win-Digipet (ab WDP 2021.2d)**

© Markus Herzog, Win-Digipet

### ***Einleitung***

Die hier vorgestellte Selbstbau-(neudeutsch: DIY/Do it Yourself)-Fernbedienung ist entstanden aus einem jahrelangen Hirngespinnst Win-Digipet abseits der kommerziellen Handregler/Smartphones mit einem einfachen Handregler fernsteuern zu können. Hinzu kam der Gedanke, dass die noch heute unterstützte Rangierfernsteuerung einzelner Lokomotiven via Gamepad ggf. irgendwann nicht mehr möglich sein könnte, da ein Teil der genutzten Produkte bereits heute nicht mehr am Markt verfügbar sind.

Ein weiterer Antrieb sind die vielen Gespräche auf Messen, bei Stammtischen oder Seminaren, wo immer mehr der Eindruck entsteht, dass man immer mehr kauft und immer weniger „bastelt“ und irgendwie gehört doch zur Modellbahn auf das Basteln/die kleinen Erfindungen.

Das vorgestellte Projekt soll in keinsten Weise Konkurrenz zu kommerziellen Produkten sein, sondern vielmehr dazu anregen ein bisschen selber zu basteln. Und das vor Allem für kleines Geld. Die Kosten sind so überschaubar, dass man ohne Probleme jedem (Besucher)Kind/Enkel etc. einen solchen Handregler in die Hand drücken kann.

Die Schaltung ist bewusst sehr simpel und hoffentlich gut nachvollziehbar gestaltet. Natürlich geht das auch alles komplizierter mit Display etc.. Wer sich ein bisschen mit dem Code beschäftigt kann da bei Bedarf sicher für sich selber noch nachlegen.

### ***Und dann noch das Übliche...***

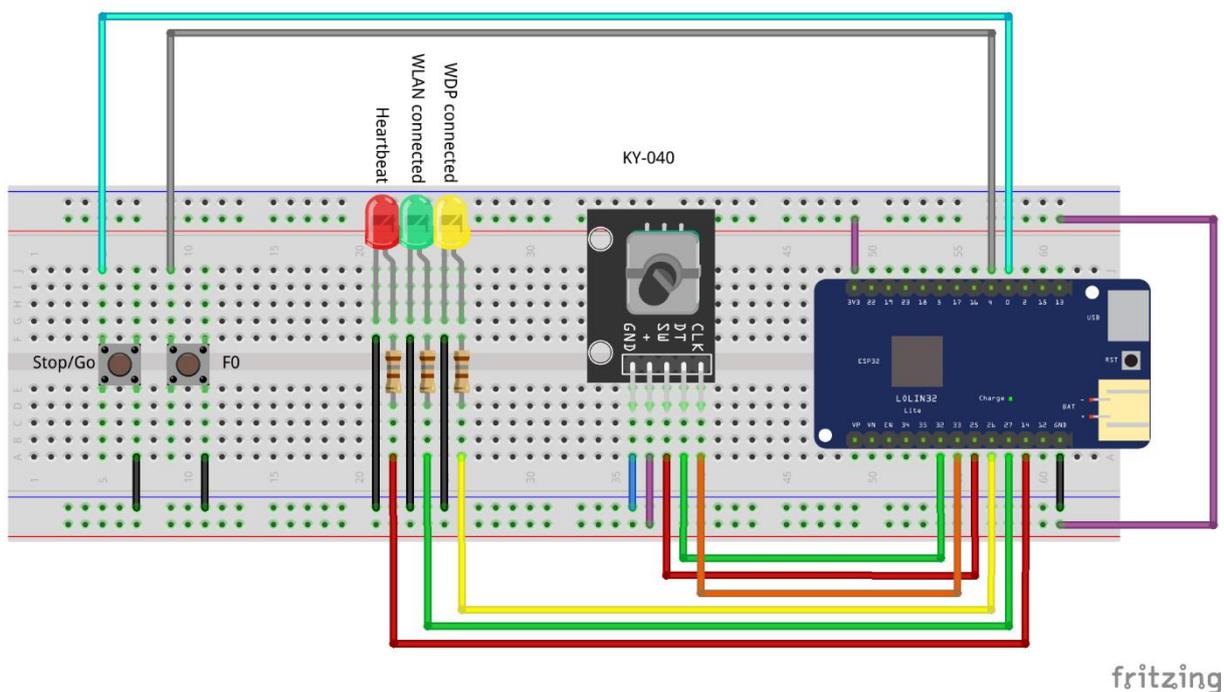
Der Nachbau der Schaltung und jegliche Weiterentwicklungen geschieht natürlich auf eigene Gefahr. Insbesondere der eventuelle Einsatz einer Lithium-Batterie hat bei falscher Handhabung entsprechende Risiken. Des Weiteren raten vom Bauteil-Bezug auf „dubiosen“ Online-Quellen ab. Beim Bezug über einen bekannten Online-Buchhändler raten wir dazu die Rezensionen vorab zu Rate zu ziehen.

## Die Schaltung

Die Beispiel-Schaltung wurde auf einem sogenannten Bread-Board aufgebaut wie man es gerne für Testprojekte nutzt. Natürlich wäre das Ziel die eigene Schaltung in einem kleinen Plastikgehäuse im Stil der bekannten Loconet FRED unterzubringen.

Die Bauteile sind alle in gängigen Online Geschäften wie Reichelt oder Amazon ohne Probleme zu bekommen. Aufgebaut ist das Gerät basierend auf einem ESP32 Lolin Lite. Es gab aber auch nahezu jeder andere ESP32 verwendet werden. Vorteil an dem ESP32 Lolin Lite ist die Möglichkeit eine Lilon Batterie als Spannungsversorgung nehmen zu können (einfach Schalter dann in Batterieleitung einbauen). Es ginge aber auch ein „normaler“ ESP32 mit einer Powerbank zum Beispiel. Source-Code und Schaltplan werden kostenfrei zur Verfügung gestellt. Die stets aktuellste Source-Code-Version mit Änderungshistorie ist zu finden auf:

<https://github.com/mherzog3/ESP32-WDP-Remote-Simple>



Die Demo-Schaltung enthält eine Taste für Stop/Go und eine Taste für F0. Des Weiteren enthält die Schaltung 3 LED mit den Funktionen (alle Vorwiderstände für die LEDs habe einen Wert von 180Ω):

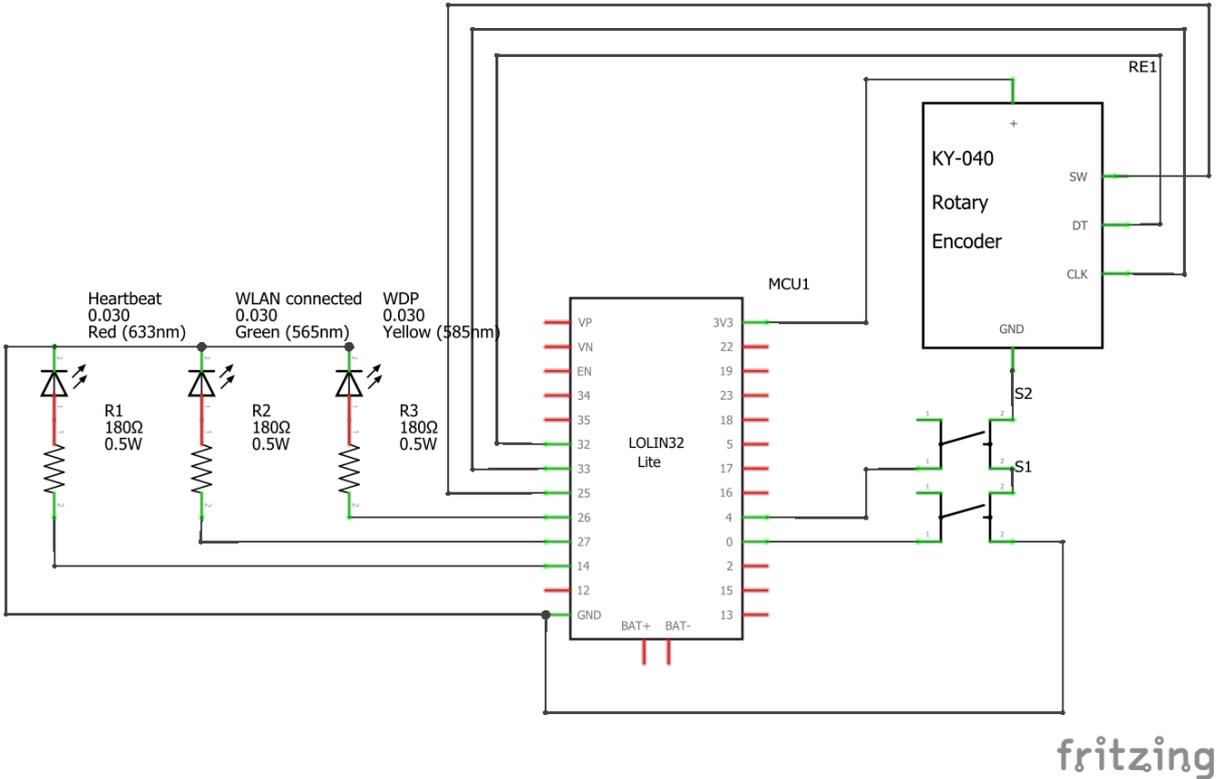
- rot: flackert ähnlich einem Herzschlag/Heartbeat, Anzeige „ich lebe“
- grün: WLAN verbunden
- gelb: Verbindung zu WDP hergestellt

Der verwendete Inkrementaldrehgeber dient zur Geschwindigkeitsregulierung der verwendeten Lok und durch Druck auf den Drehregler kann man die Richtung ändern.

Man sollte darauf achten einen KY-040 mit der kleinen Platine zu kaufen, sonst muss man den weitergehend beschalten.

Weitere Tasten/Anzeige lassen sich ergänzen. Der Mikrocontroller hat ja noch eine Menge Anschlüsse frei. Der Code kann damit umgehen, muss man nur die Config.h entsprechend anpassen. Allerdings muss man schon sich mal minimal mit dem ESP32 befassen, denn nicht alle Anschlüsse sind frei nutzbar. Es gibt z.B. welche die nur als Ein- oder Ausgang gehen. Andere brauchen evtl. ein Pullup (34,35,36,39 z.B. beim Lolin Lite). Am besten sucht man im Netz nach "ESP32 Lolin Lite Pins To Use".

Hier auch noch der Schaltplan:



## **Source-Code und Kompilieren**

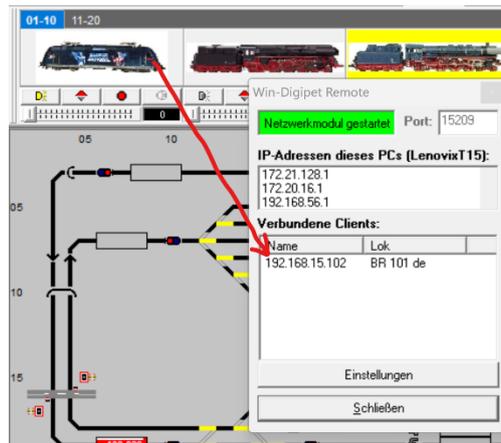
Kompiliert und auf den Mikrocontroller via USB-Kabel aufgespielt werden kann das Programm mit der Arduino IDE, die man kostenlos im Netz erhält. Einfach die „ESP32-WDP-Remote-Simple-Demo.ino“ und die „example-config.h“ in ein Verzeichnis namens „ESP32-WDP-Remote-Simple-Demo“ entpacken. Dann die „example-config.h“ in „config.h“ umbenennen und die ESP32-WDP-Remote-Simple-Demo.ino aus der Arduino IDE heraus aufrufen. Hier muss man ggf. noch den ESP32 zur Board-Bibliothek hinzufügen als auch die Bibliothek „AiEsp32RotaryEncoder“. Auch zu diesen Schritten gibt es zahlreiche Anleitungen im Netz. Wir beschreiben dies auch ganz bewusst nicht in kleinsten Schritten in diesem Dokument, da es erstens im Netz schon hunderte Male niedergeschrieben wurde und wir zweitens auch zum Basteln/Experimentieren anregen wollen.

Der Source-Code ist in weiten Teilen kommentiert und kann ohne Probleme auf eigene Wünsche angepasst werden. Die Kommentierung ist bewusst in Englisch, sollte aber einfach verständlich sein. Der Source-Code soll auch keinen „Schön-Programmieren“ Preis gewinnen, sondern es wurde vor Allem Wert auf Verständlichkeit gelegt und nicht auf höchste Programmierkunst. Programmteile ohne umfangreiche Kommentierungen sollten ohne tieferes Programmierverständnis erstmal nicht geändert werden 😊

So ließen sich ohne weiteres z.B. weitere Funktionstasten oder Statusanzeigen ergänzen. Alle Konfigurationsanpassungen finden in der config.h statt. Für den ersten Start (vorausgesetzt die Schaltung wurde 1:1 aufgebaut) reicht es erstmal dort den eigenen WLAN-Namen (SSID) und das WLAN-Passwort zu ergänzen (WPA2-Netzwerke werden unterstützt). Außerdem muss noch die IP-Adresse des WDP-Rechners eingetragen werden. Im Zweifelsfall kann man auch erstmal grundsätzlich die Schnittstelle programmieren in dem man die WDP Mobile App auf seinem Android oder iOS-Smartphone installiert und sich mit WDP auch über das eigene WLAN verbindet.

## Lok-Zuweisung in WDP

Das geschieht auch ganz einfach. Man öffnet das WDP-Remote-Fenster in WDP (grünes Antennensymbol in Symbolleiste) dort sollte nun nach erfolgreichem Verbindungsaufbau die IP der Selbstbaufernbedienung erscheinen. Einfach nun aus der Lokleiste wie als wenn man eine Lok auf ein FAZ ziehen wollte diese auf die IP-Adresse des Handreglers in der Liste ziehen und loslassen. Und schon sollte sich die Lok via Drehgeber steuern lassen können.



## Beispiel-Realisierung

Das nachfolgende Bild zeigt wie ein einfacher WDP-Handregler im Gehäuse aussehen könnte basierend auf der obigen Demoschaltung (geht sicher schöner 😊):

