

Die Liga

Ein Arbeitskollege hat eine auf der [Elo-Zahl](#) basierende Tischtennis-Liga Webseite gebaut. Diese verwendet SocketIO in JavaScript und war an sich schon sehr komfortabel zu bedienen. Mir selbst und ein paar anderen fiel es aber schwer sich in der Pause die gemachten Spielstände zu merken um sie dann später auf der Webseite einzugeben.

Die Idee

Ich hatte dann die Idee einen batteriebetriebenen Apparat zu bauen wo man noch vor Ort nach jedem gespielten Satz die Punkte direkt und einfach eingeben konnte. Da wir im Bürogebäude jeder einen Schlüsselchip haben um in das Büro zu kommen, kam die Idee auf diesen auch für das Erkennen des aktuellen Spielers zu benutzen.

Es sollten also beide Spieler durch den Chip erkannt werden und man sollte Zahlen eingeben können. Eine Art Feedback brauchte man auch, also ein kleines Display war auch nötig. Aus den Anforderungen ergaben sich auch schon die verwendeten Komponenten ...

Die Hardware

Verwendet wurden:

- Raspberry Pi Zero W ([amazon](#))
- RC522 RFID Reader für SPI ([amazon](#))

Raspberry Pi Zero Tischtennis Terminal

Geschrieben von: jjk

- 20x4 Zeichen LC Display ([amazon china](#) , [amazon de](#))
- USB Keyboard Tastatur ([amazon](#))
- Micro USB OTG Adapter ([amazon](#))
- Ein paar Litzen aus einem alten IDE Kabel

Die Stromversorgung übernimmt eine Powerbank, mit dieser von Anker hält der Terminal 4 Tage Nutzung aus: [Anker PowerCore 20100](#) . Aber es muss natürlich nicht so eine große Powerbank sein. Der ganze Terminal verbraucht so 150-180mA im Betrieb.

Die Software

Da die vorhandene Webseite eine JavaScript App mit SocketIO ist, sollte der Terminal die Kommunikation nach Möglichkeit auch über Socket machen damit am Server Code nicht zu viel geändert werden sollte. Die Software ist unter https://github.com/thedarkman/ttleagueter_minal zu finden. Sie ist in Python geschrieben und läuft auf dem Raspberry Pi Zero quasi in Endlosschleife. Nachdem man den Pi mit rasbian ausgestattet hat folgen die üblichen Schritte wie Dateisystem vergrößern, Wlan konfigurieren, SSH aktivieren (sehr nützlich), ...

Ich hatte diese initialen Schritte noch mit einer normalen Tastatur am OTG Adapter und einem angeschlossenen Monitor erledigt.

{!image}attl_terminal/attl_terminal_01.jpg{/!image}

Sobald aber das Wlan konfiguriert ist lief alles nur noch über ssh. Man braucht ein paar Zusatz Pakete auf dem Pi wenn man, wie ich, das minimal Image aufgespielt hat. Neben Git sind das noch python-dev und diverse andere die auf der [github Seite](#) genannt sind.

Der RFID Teil

Wenn man nach "Raspberry Pi RFID" bei google sucht, findet man schon viel, ich habe mich an die [T4rffner Seite](#) gehalten. Dort fand ich auch den Link zur Python Klasse MFRC522.py (Source bei [github](#)) die ich allerdings später noch patchen musste. Die ersten Versuche fanden noch mit Steckbrett Verkabelung statt. Hat aber sofort funktioniert und ist echt schnell. Danach wurde auch das Display dazu geholt und beides fest verläftet (mit Steckern an den "Geräten")

Zu diesem Zeitpunkt lief das Python Programm nicht mehr, da die Display Bibliothek und die R C522 Bibliothek verschiedene GPIO Modes nutzen. Dies war schnell in der MFRC522.py korrigiert.

{limage}attl_terminal/attl_terminal_02.jpg{/limage}

Das LC Display

Zuerst war nur ein 16x2 Display geplant, aber das war einfach zu wenig Platz für die Informationen die man bei den Spielen anzeigen wollte. Mit einem Austauschdisplay in 20 Zeichen und 4 Zeilen ging es dann an den ersten Gehäuse Prototypen

{limage}attl_terminal/attl_terminal_03.jpg{/limage} {limage}attl_terminal/attl_terminal_04.jpg{/limage}
age} {limage}attl_terminal/attl_terminal_05.jpg{/limage}
g{/limage}

Mittlerweile wurde das neue Display geliefert. Blaues Display mit weißer Schrift, sieht man viel besser als das alte unbeleuchtete.

Der Spielablauf

Beim Anlegen der Spannung wird das Display noch nicht initialisiert und es zeigt einfach weiße Balken

{limage}attl_terminal/attl_terminal_06.jpg{/limage}

Sobald der Python Code läuft und das Display initialisiert, wird die aktuelle IP zusammen mit dem Startbildschirm angezeigt

Raspberry Pi Zero Tischtennis Terminal

Geschrieben von: jjk

{limage}attl_terminal/attl_terminal_07.jpg{/limage}

Danach muss einmal mit Enter bestätigt werden, dies war nötig um den System-Keyboard-Puffer zu leeren. Danach wird versucht sich zur Server-App zu verbinden ...

{limage}attl_terminal/attl_terminal_08.jpg{/limage} {limage}attl_terminal/attl_terminal_09.jpg{/limage}

War die Verbindung erfolgreich, so geht der Terminal in seine "Arbeitsschleife" und wartet darauf dass die RFID Chips gescannt werden.

{limage}attl_terminal/attl_terminal_10.jpg{/limage}

Wird ein RFID Chip gescannt wird ein Request an den Server gesendet und nach dem Spieler gesucht und wenn dies erfolgreich war, der name und die aktuelle Elo Zahl des Spielers angezeigt

{limage}attl_terminal/attl_terminal_11.jpg{/limage}

{limage}attl_terminal/attl_terminal_12.jpg{/limage} {limage}attl_terminal/attl_terminal_13.jpg{/limage}

In diesem Spiel spielen also Simon und Mark gegeneinander.

{limage}attl_terminal/attl_terminal_14.jpg{/limage}

Raspberry Pi Zero Tischtennis Terminal

Geschrieben von: jjk

Nachdem nun das Spiel angelegt wurde zeigt sich der Eingabe Bildschirm in dem mit blinkendem Cursor auf die Spielstände des aktuellen Satzes gewartet wird. Dort sieht man auch die möglichen Änderungen in der Elo Zahl falls Spieler eins gegen Spieler 2 gewinnt (oberer Wert, bei uns hier +13) und wieviel er verlieren würde wenn er den Satz verliert (untere Zahl, hier -19)

{limage}attl_terminal/attl_terminal_15.jpg{/limage}

Ist ein Satz zuende, gibt einer der Spieler den Spielstand mithilfe der Tastatur ein und bestätigt jeden Wert mit der Enter-Taste, oder löscht eine Zahl mit Backspace

{limage}attl_terminal/attl_terminal_16.jpg{/limage} {limage}attl_terminal/attl_terminal_17.jpg{/limage}

{limage}attl_terminal/attl_terminal_18.jpg{/limage} {limage}attl_terminal/attl_terminal_19.jpg{/limage}

Jeder eingegebene Satz wird in den letzten zwei Zeilen des Displays angezeigt. Werden keine 5 Sätze gespielt, wie hier, gibt man einfach zweimal Enter ein und es erscheint eine Zusammenfassung des Spiels während es zum Server gesendet wird

{limage}attl_terminal/attl_terminal_20.jpg{/limage}

Ist das Spiel am Server verarbeitet, so sendet dieser aktualisierte Daten zurück und der Terminal zeigt noch die gesamte Änderung an

{limage}attl_terminal/attl_terminal_21.jpg{/limage}

Danach geht der Terminal wieder in seinen Wartestatus für die RFID Chip Erkennung.

Es gibt eine Spezielle RFID Chip Id, konfigurierbar, wenn dieser Chip gescannt wird, führt der Raspberry Pi Zero W das System herunter. Sicher ist sicher, so kann das Dateisystem nicht so schnell kaputt gehen.

{!image}attl_terminal/attl_terminal_22.jpg{/!image}

Das Fazit

Ich habe schon vorher in Python programmiert und z.B. eine [Photobooth mit Raspberry Pi 3](#) und meiner DSLR gebaut, aber dieses Projekt ist das erste wo ich ein bisschen auf die Programmierpattern von Python geachtet habe. Auch die Themen LC Display und RFID Reader waren sehr interessant und das Gelernte wird sicher auch anderweitig Anwendung finden. Die Bedienung des Terminals hat bis jetzt jeder hinbekommen und ist relativ fehlerunanfällig. Ich musste allerdings die Möglichkeit einbauen eingegebene Zahlen auch wieder zu löschen. Alles in Allem hat sich das Projekt gelohnt und ich werde nun ein schönes Gehäuse planen. Die Ferrero Plastikverpackung ist zwar ok, sieht aber nicht unbedingt schön aus. Sie erfüllt ihren Zweck die Elektronik zu schützen. Mehr aber auch nicht.