The League

A collegue of mine has developed a website for our internal company tabletennis league that is based on the elo rating system . His website uses socketIO and ist quite nice to use. But for myself and some other guys it was a little bit difficult to remember all set values if we played in our free time.

The Idea

So an idea came to my mind that we have a battery pwoered device at the table tennis field to send the points directly to the server. The fact that each of us has his own rfid tag at the keyring just made it clear to use them as an identification token for the player.

Both player should be identified by the rfid tag and we need to be able to insert numbers. To show some feadback i added a small character display. So from these requirements the hardware was set ...

The Hardware

I used the following components:

- Raspberry Pi Zero W (amazon)
- RC522 RFID Reader for SPI (amazon)
- 20x4 character LC Display (amazon china , amazon de)
- USB Keybad Keyboard (amazon)
- Micro USB OTG Adapter (amazon)
- Some different length of wire

Power will be provided from an USB powerbank. As the complete device only consumes about 150 - 180 mA i calculated about 4 days of running.**The Software**

As the already exsisting website uses socketIO, the terminal should use this too so we don't need to change much at the server side code. So i developed the software for the terminal in python and it is available here: https://github.com/thedarkman/ttleagueterminal .

The code is running on the Raspberry Pi Zero in an endless loop. To setup the Pi you will have to do all the common steps like expanding the filesystem (newest version of rasbian does it

automatically), configure WiFi and activating the SSH daemon

To be able to complete the initial setup steps, unless ssh is activated, i used a normal Monitor and a small USB keyboard for this. After ssh was active i only used this. And then we can even disable HDMI and video output. This is a picture of what it looked like with all the adapters to get the Pi Zero connected (also the rfid reader is already connected)

{limage}attl_terminal/attl_terminal_01.jpg{/limage}

I needed some additional packages on the Pi to be installed because i'm using the minimal image of rasbian for this project. I installed git, python-dev and some other dependencies that are mentioned on the github page .

The RFID stuff

If you google for "Raspberry Pi RFID" you will find a lot of informations how to connect the rfid reader to a Pi. I found also a python class MFRC522.py (source on github) that i could use, but i had to patch it later. The first tests were done on breadboard and scanning rfid chips and cards worked really good and fast. After that i connected the 20x4 lc display to the Pi and also permanently attached the rfid reader (leaving headers to remove the both devices)

At this moment my python code did not work anymore because the libraries for the display and for the rfid reader uses different GPIO modes. I patched the MFRC522.py file because it only uses one gpio pin for the reset signal. After that all worked fine.

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

The LCD

In the first place i planned on using a 16x2 display, but that was far to less space to show all the informations i wanted to show during a match. With a temporal 20x4 display i startet building a housing prototype from an old sweets package.

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

The new display arrived very quick and now with the blue backlight and white text color display it's much better.

The Match

When powering the device (terminal) there is nothing to see yet, just the initial blocks in the display.

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

As soon as the python code is running the welcome screen appears and shows also the current ip address (useful if you need to connect via ssh to the Pi).

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

After that you have to hit enter and the terminal tries to connect to the server app ...

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

If the connection was established successfully the terminal starts it's working loop and waits for the rfid tags to be scanned.

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

When scanning a rfid tag the player is requested from the server with this rfid tag-id and if a player was found the server replies with the player's name and current elo value

{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 this match we will see Simon playing against Mark

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

After the creation of the match the input screen appears and waits with a blinking cursor for the set points to be entered. This screen also shows the possible elo change values if player 1 beats player 2, in this case here +13 and if he loose he will loose 19 points.

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

If a set is finisched one of the players can enter the points. This could be done with the keypad and each set point has to be confirmed with enter or can be corrected using 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}

Each set is shown in the last two lines of the display. If the players don't play 5 sets, as it was the case in this match, they can complete the match with entering enter two times. After that a match overview is shown while the match is submitted to the server.

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

If the match was processed at the server side, the server sends updated data back and the terminal shows the real elo changes for a few seconds.

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

After that the terminal restarts it's loop and waits for rfid tags again.

But there is one special rfid tag-id that will cause the Pi to shutdown and say goodbye. This was added because of the filesystem of the Pi. As it is a linux system the best way to switch of such a system is a clean shutdown.

{limage}attl_terminal/attl_terminal_22.jpg{/limage}

The conclusion

This as not my first python project as i adapted a Photobooth with Raspberry Pi 3 software to my needs and my DSLR, but the first one where i tried to program with all the python programming patterns. The topics lcd and rfid are also very new to me, at least on a Pi and i guess the lessons learned will be helpfull in other situations. The useability of the terminal is ok and every player has used it well. The only thing i had to add is the "delete set points" feature. As the project was successfull i will create a nicer case for it as the prototype just protects the electronics but not more. It does not look cool ...