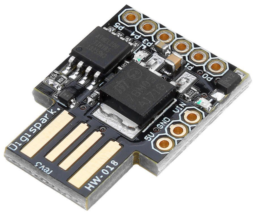
Digispark (Default) Programming Evolution:

Borya K.

Topics I improved:

- Batch basic programming
- How to start background process in Windows

What 's Digispark?



It’s a low cost alternative of the famous HID Rubber Ducky, similar to Malduino, quite cheap and relatively low tolerance to fatigue (pluggin in and out).

Programmed in arduino with their libraries, needed to add the board.

Link for set up

The Practice:

The idea:

Be able to open some programs attached and have the possibility to execute commands.

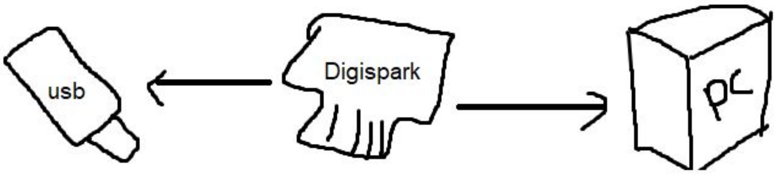
Set up:

A hub which has connected the digispark and a little usb for storing the programs.

1ºst Prototype:

At first I faced it the wrong way, I focused on using as much as possible the digispark at all cost. That concluded on a long, complicated and tricky single program. As it was so long, I needed to learn a way to redirect the strings to the flash memory that seemed to fix the problem.

1ºst Prototype



As we can observe, the one that constantly interacts with the PC is the digispark, which can use the usb as support for extra functions.

Program:

As we have talked before, you can now see the program here and see their problems.

Pros:

-First contact with the device

-Learned the basics of its libraries

Cons:

-As it worked all by entering strings and waiting a fix amount of time, if the shell (for example) had booted a bit slower than usual, that caused the whole program to fail and needed to repeat.

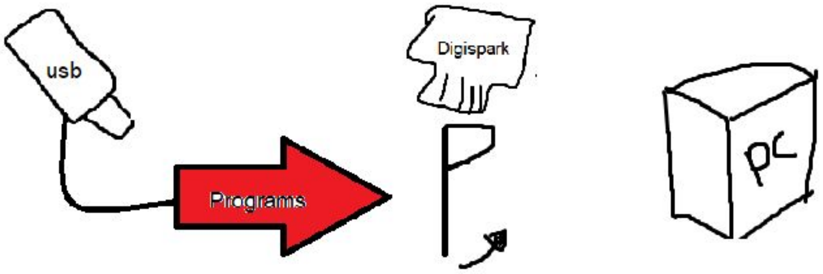
-Really messy and not so discrete.

2ºnd Prototype: (Final [At 1/12/2020]) :

At this time, I decided to face it from a different perspective.

This time, digispark will only be the detonator, a trigger that will execute a Batch program (.bat).

2nd Prototype



As we can see here, digispark plays the role of a simple trigger.

By doing this, we get the following **pros**:

- We don't overload the device with unnecessary data. Making the program far more simple.

-Much more discrete, the detonator will be "Win + R" inside of it we'll write "Powershell [command]".

-Easier to implement new functions, you won't have to always modify the source code of the digispark. Just modify the .bat file.

-The .bat file can execute plenty of options, for example we could start a python program.

-And the most important. We completely erase the problem of timing. As the batch file is a linear consecutive program. We don't have to worry about it.

Cons:

-We need to stay fix with names and paths of the files related to the USB.

Program:
The digispark program is probably the simplest part, the other two files are the most important ones . There's a little test if you want to check, simulating "That you've been owned", with the excuse of practising how to boot browsers with determined links or simple os libraries of python.
CAREFUL!! By default, python won't shutdown the PC, cause it is modified as a commentary but you can test it if you prefer.

The program

Conclusion:

The last method is quite more effective and has much more potential, we can clone repositories and execute different tools and external programs. Even though carrying it in a usb is more secure.

I've learned a lot, mainly of cmd and powershell syntax and functioning.
I don't know if this can be put a bit farther by implementing something like a web server with an ESP8266 or something similar like in some versions of malduino.