Name class school

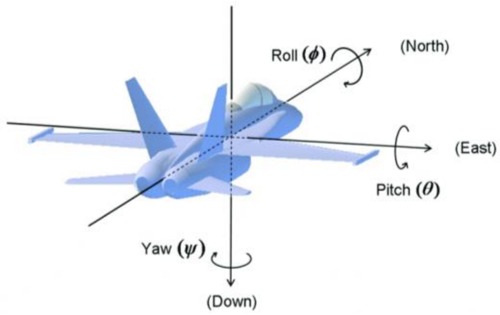
5.10.2023 - 23.05.2024

Yearly project: HO(TA)S

Hands on (Throttle and) Stick

Contents

Projevtoverview



A **H**ands **O**n **T**hrottle **A**nd **S**tick (Image to the right) is a computer input device that visually resembles the control surfaces that can be found in an airplane/helicopter. It is mainly used to learn and train the flying of real aircraft in simulators, and for some years now they have also been used in the entertainment sector to control flying objects (airplanes, helicopters, spaceships, etc.) in computer games and simulations such as DCS, WarThunder, etc. The main module of the HOTAS, the flightstick or joystick, performs the main functions of “roll” and “pitch” (illustration on the right). The fact that these axes are corrected analogously in a HOTAS means that much greater precision can be achieved than would ever be possible with a keyboard. There are also buttons on the stick for entering other commands.

Main Goal

The aim of the project was to work on and complete a project that was started outside of school. It should consist of two Arduinos, one of which is in the base and another in the joystick. These two µCs are to exchange the required data via I2C and pass it on to the PC via the USB interface. In addition to the obvious advantages of having more pins available, the choice of two Arduinos also offers the advantage of being able to add further modules via the I2C bus using the Arduino in the base, with just a few lines of code.

Optional Goals

Optional 1

For the time being, the addition of a further module was planned as an optional target. This module would be the throttle. It is usually located to the left of the pilot in real aircraft and allows stepless control of the thrust in airplanes or the pitch of the rotors in helicopters as well as control over the speed of the turbine.

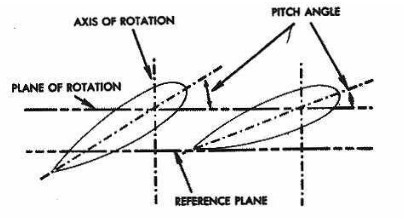


FIGURE 4.—The pitch angle of a rotor blade is the angle between the chord line and a reference plane determined by the rotor hub or the plane of rotation.

Optional 2

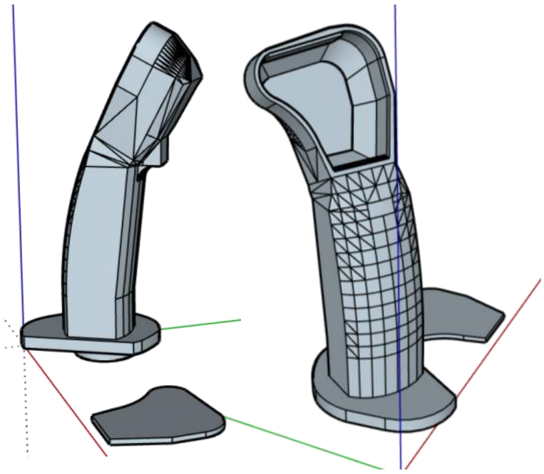
Another option was the output of data, which could be retrieved from the simulation via third-party software, on digital or analog displays mounted near the joystick. However, this goal was only a rough idea from the beginning and was never really started. However, it would be a good placeholder if there was still a lot of time left at the end of the year. However, this was not the case and so this optional goal was already discarded at the end of January.

Implementation

Project planning

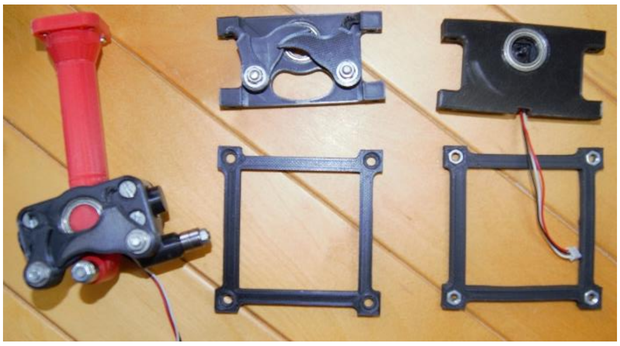
Research 3D model

The project was started with research on different joystick models to get a feeling for what is needed on a joystick. After some first own drawing attempts (pictures on the right), it was decided to use an open source 3D model of a F-16 / A10 joystick. As it turned out that it would take a lot of time to draw it completely by oneself. It was decided to use the 3D model from JFlyer81, as it is based on a 3D scan of a real joystick.



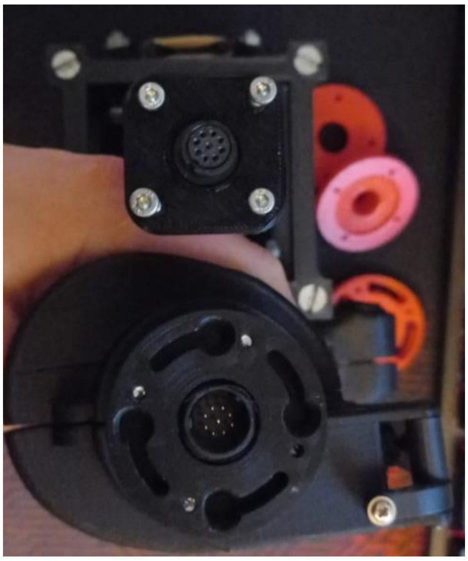
However, since the final joystick consists not only of a joystick, but also a part for mounting the joystick, the so-called gimbal. The basic idea here was a construct that could rotate around two axes. During the first attempts to design our own gimbal, we quickly realized that it was very difficult to draw a concept that would keep the force that counteracts the movement of the joystick linear and prevent it from noticeably “snapping” in the middle position, which would be optimal for a joystick. After lengthy research, a design for a gimbal that roughly met the requirements was found (olukelo >> Thingiverse). This gimbal was then used as the basis and further modified.

3D drawing/adjustment



1 Individual components of the gimbal are recognizable ball bearings and potentiometers

These modifications were mainly related to the adjustment of the exploitation method from Hall sensors to poorer/less precise potentiometers and various smaller adjustments concerning ball bearings, cable ducts and mounting points for nuts etc.. The next big and important adjustment was the modification of the joystick shaft which firmly connects the gimbal and joystick. The original connection was a pure plug-in connection which relied purely on the friction of the materials. There was also a connection option based on two M5 screws, but I also found this unsuitable as I wanted the joystick to be able to be replaced quickly if necessary.



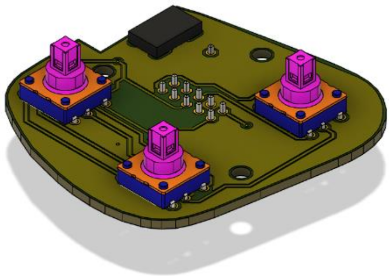
2 "Twist on quickconect", gimbal top, joystick bottom

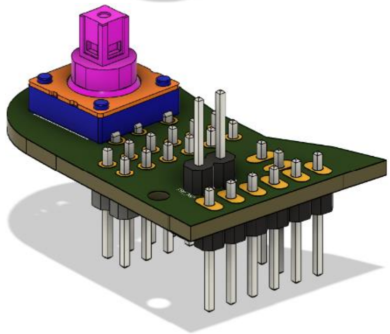
As a result, I redesigned the connection mechanism into a variant that locks. The joystick is pressed onto the connector with a 45° turn to the base and turned back to a 0° position, whereby the joystick locks with the shaft and is firmly connected to it. After an unsuccessful attempt with a makeshift circuit board, the electronic connection is provided by a push-in connector, which is normally used for audio systems in the avionics sector (see picture on the left). After the basic model of the complete joystick assembly was ready, the planning of the electrical components was started.

Electronic components

The basic idea here was to have an independent Arduino in the joystick which takes care of reading out the various buttons and sending them to another Arduino which then takes care of sending this data to the PC. Reading the 26 buttons in the joystick should still prove to be a hurdle, but more on that later. In the base of the joystick is an Arduino Micro, which has the properties to appear as a keyboard/mouse/controller under Windows. This reads the data sent from the Arduino Nano and passes it on, and also passes on the values of the potentiometers.

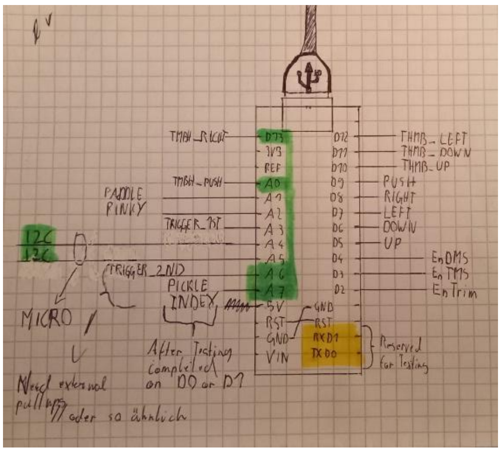
The potentiometers used here were potentiometers which had a rotation range of approx. 90°, instead of the classic 300°-320°, which corresponded almost exactly to the possible rotation angle of the gimbal.





In the first prototype, the buttons were small holders for five 5*5*3 mm buttons attached to the housing with screws. However, this structure was very complex, unreliable and prone to errors. It was therefore decided to replace these brackets with an integrated 5-way button, which, however, requires a circuit board. The next step was therefore to design one or two circuit boards that corresponded to the joystick housing, a cut was made in the housing so that the circuit boards could be firmly installed in the housing. This process was repeated for the side circuit board and continued accordingly. As there is a front board and a thumb board, which are in different positions (on the left where the thumb is and on top of the front panel).

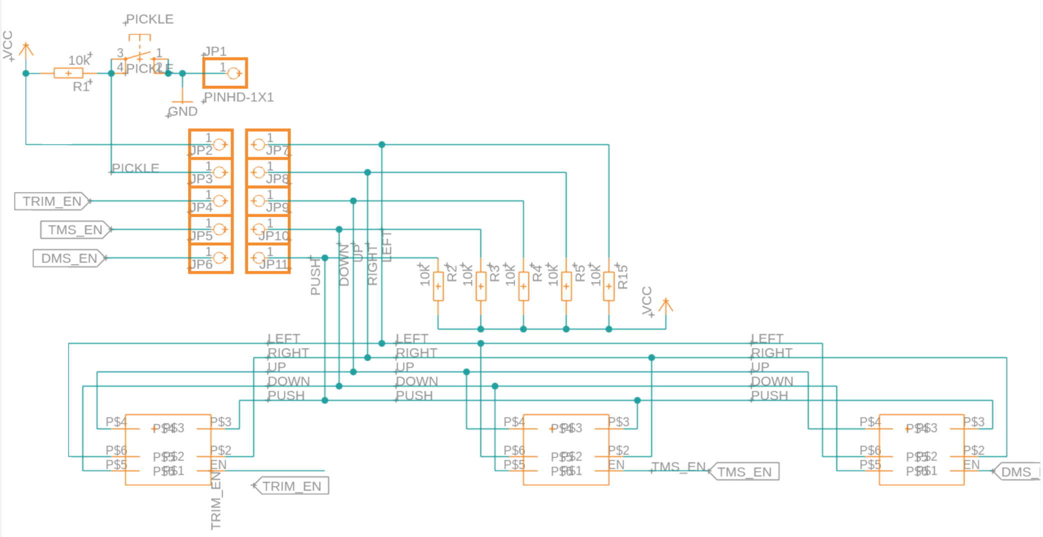
Electronic implementation



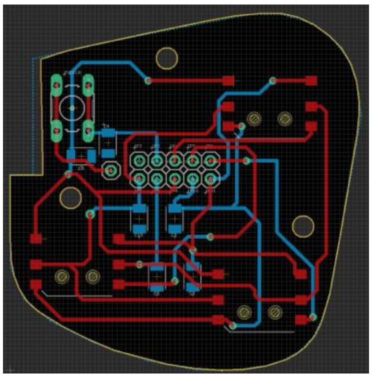
3 Pinout/use of pins => Nano mounted in the stick itself

As the Arduino Nano only has 14 digital and 8 analog pins, two of which are lost for the communication between the two Arduinos via I2C, only 20 pins remain. As already mentioned, there are 26 buttons in the joystick. However, since the 5 way buttons have a common supply pin, they can be switched on and off sequentially and the 5 different lines of the respective pins can be combined with the corresponding ones. This means that the 15 pins that would have to be used for this can be reduced to 5 pins for reading out the statuses and 3 pins for switching the button units on and off. For the other buttons, these were not solved using the same system, but were pulled individually onto the µC, as there were now enough pins available and the wiring would not have been much easier.

Schematic front PCB:

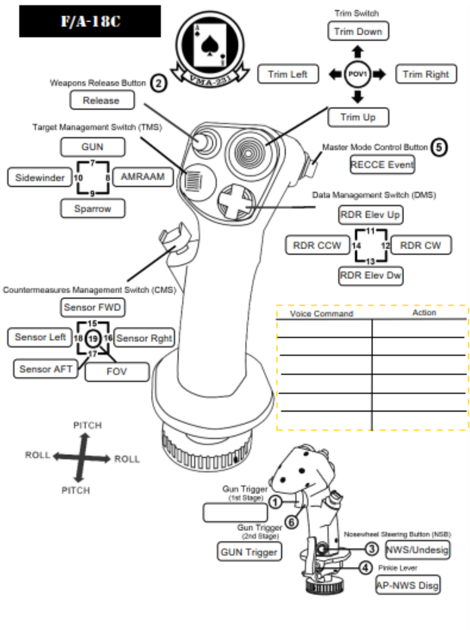


4 Schematic front PCB

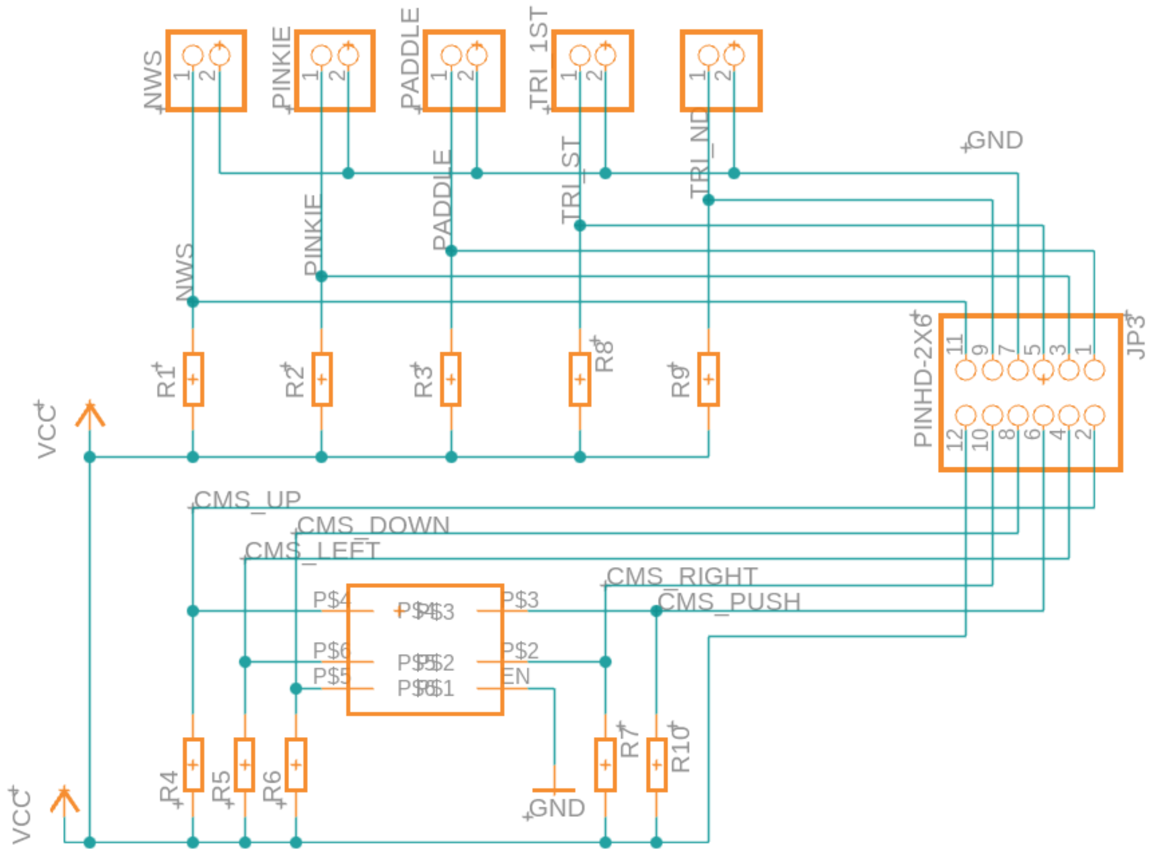


5 Related PCB

Here you can clearly see how the outputs of the various button units all go to the same five conductors. It should be noted that since the pins are all configured to pull-ups, only the unit that is currently switched to GND by the µC can control the outputs. It should also be mentioned here that the names TRIM, TMS, DMS refer to control characteristics of the airplane, I refer to the illustration on the right, and these have only been adopted for orientation purposes.

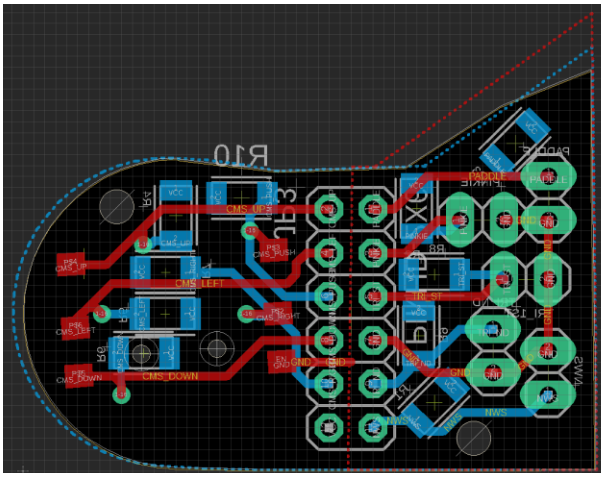


Schematic thumb PCB:

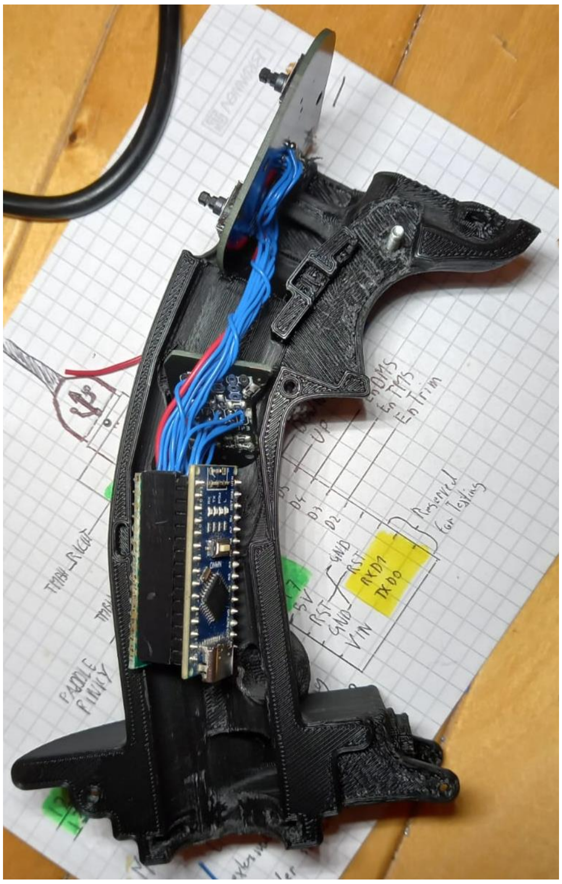


7 Schematic thumb board

The board installed on the side for the Daument 5Way also served as a distribution board so that not every other button that was otherwise installed needed its own GND cable and so in the end only three GND cables went out from the Nano, one to the front board, one to the side board and one to the base.



6 Board



8 Joystick during soldering

Wiring

In practice, the cabling no longer looked so neat and tidy, as everything had to fit into a very limited space. On the right you can see that, despite the reduction to 10 cables, it became a little confusing.

Software

The software running on the Arduinos was developed at the same time as the setup was being planned. If you start at the base, there is a library on the Arduino Micro which ensures that the Micro appears as an HID. This is then extended with the necessary joystick code so that the Micro can work or be recognized as a joystick.

The Micro writes the data read out via the potentiometers to the respective X/Y axes and accepts the incoming data from the buttons via I2C and passes it on to the PC.

The nano in the joystick assumes the underlying role here and sends the states of the buttons via I2C on request from the micro. This communication takes place via the Arduino's own I2C library.

9 Joystick during soldering

Finalization and testing phase

The actual test phase of this project was always continuous alongside the development, as a first prototype already existed. This provided a great starting point, as you could switch from the digital to the physical while planning something and get a good idea of what still needed to be changed and adapted. During the finalization phase, something unfortunately went wrong with the delivery of the circuit boards via DHL, which meant that the boards only arrived a week late, which limited the schedule somewhat but was not worth mentioning until the end. Once all the parts were on site, the assembly could begin and after about two to three afternoons the final assembly was completed. The code that had already been written in March was then adapted a little and loaded onto the controllers. After a few small adjustments (filing, soldering), the final prototype was ready and the protocol was also finalized.

10 Bild aus der finalen Testphase

Sources:

OpenSource Code

used Joystick-Lib

GitHub for this project

3D-Models

Stick Base unedited

Gimbal unedited

GitHub edited stl s

Images reference etc

https://shop.thrustmaster.com/it_it/hotas-warthog.html
https://www.researchgate.net/figure/An-illustration-of-the-three-angles-yaw-pitch-and-roll-returned-by-the-Hyper-IMU_fig1_329603549

https://www.digitalcombatsimulator.com/upload/iblock/49a/HOTAS.png

- You should also be able to find images on the GitHub.

Legal or smth I aint no lawyer

If you own the intellectual right to any of the above mentioned things and im not allowed to use them please let me know I am pretty sure that this should al be okay but anyways im just human and probably also make mistakes.