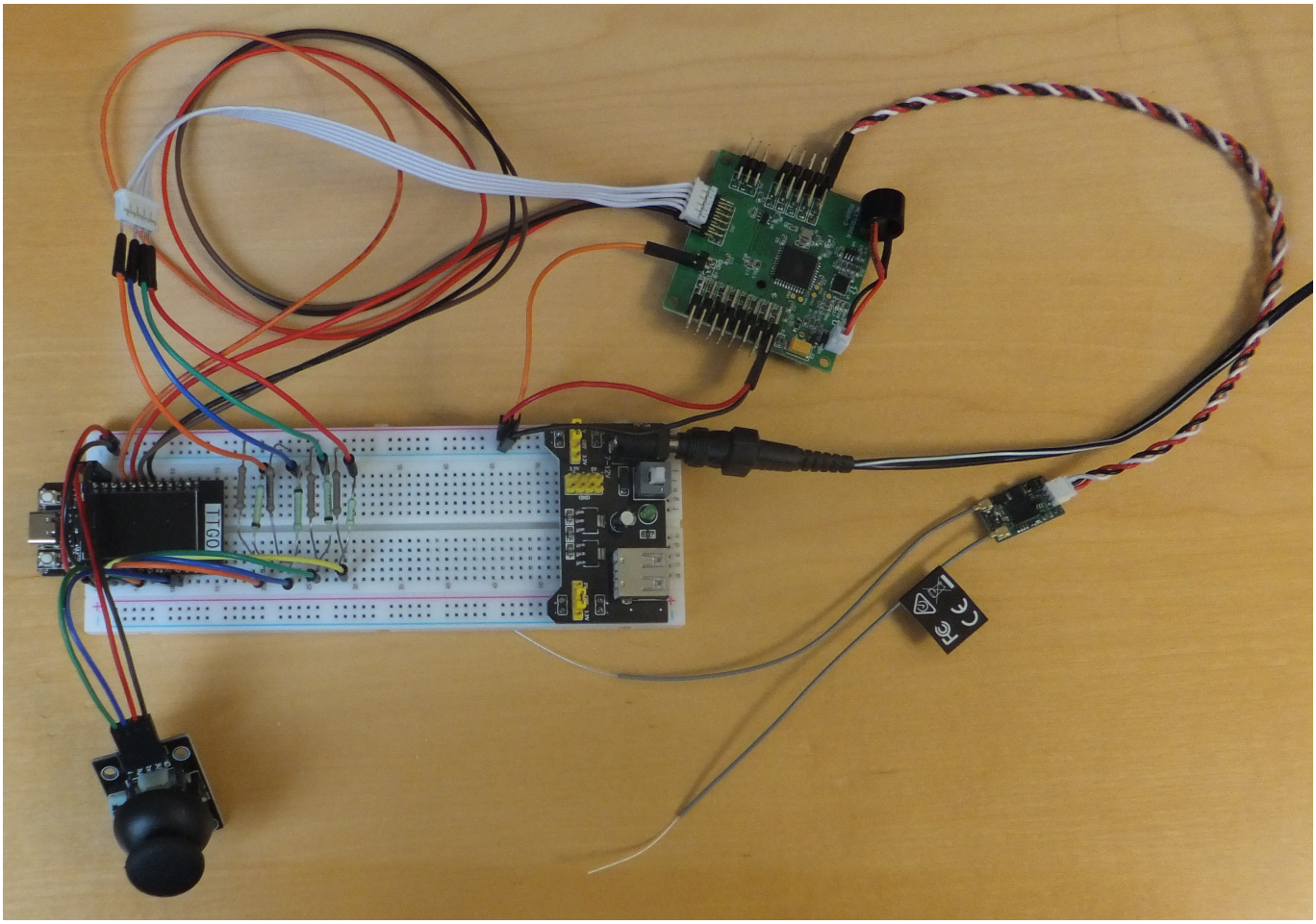
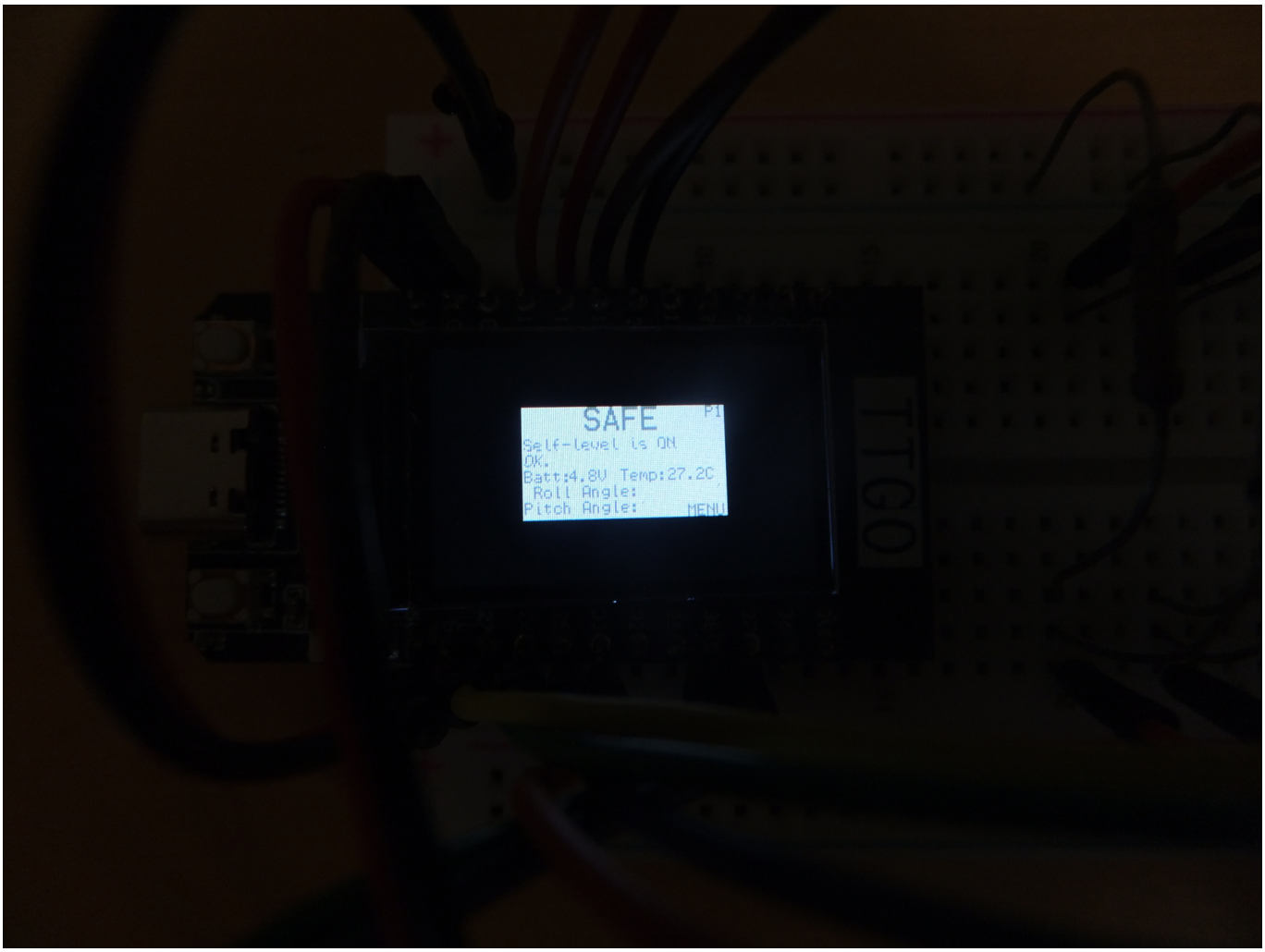
Replacement of a KK Programmer with a LilyGO TTGO T-Display

Project page on Github: https://github.com/aheyer/KK_Programmer_Replacement




SAFE
P1
Self-level is ON
OK.
Batt:4.8V Temp:27.2C
Roll Angle:
Pitch Angle:
MENU

TOC

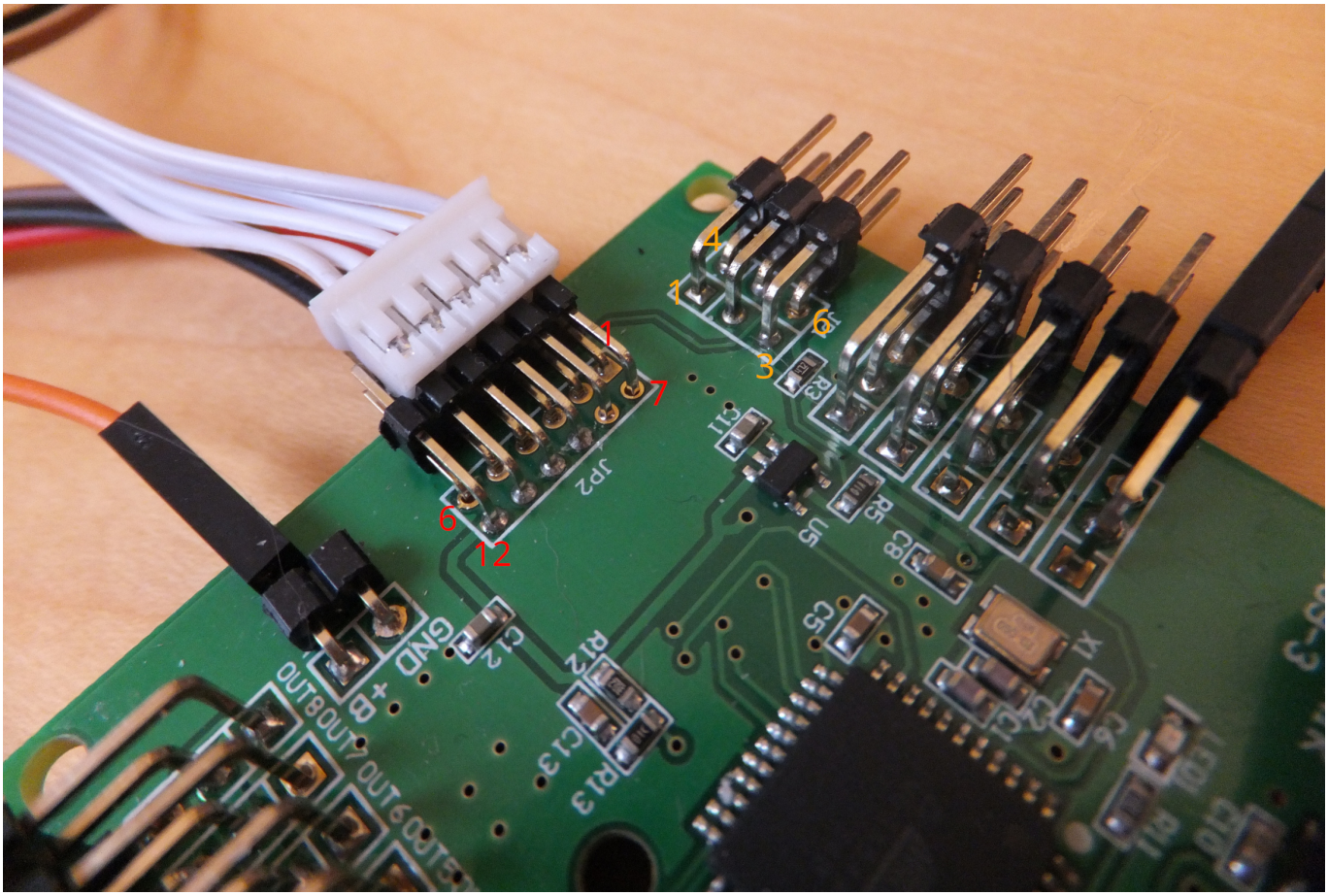
Programming a HobbyKing KK2.1HC flight controller

With the closing of HobbyKing's EU warehouse I got a couple of KK2.1HC flight controllers for $1 each. Unfortunately the needed KK Programmer device is out of stock and probably discontinued. Since the KK2.1HC lacks the display and the buttons of a regular KK2.0, this flight controller is hardly usable. I found a custom 1.6 firmware for brocken displays, that allows programming over ISP with the help of a Windows command line application. But that approach doesn't work with the latest firmware 1.19S1 Pro by Steveis. So I decided to emulate the KK Programmer with devices I found in my hobby box.

Digging into the hardware of the KK2.1HK

I wasn't able to find a picture or a description of the inside of a KK Programmer. All I had was my new KK2.1HC and the assembler code Steveis supplied with his firmware for KK2.1.x devices. Reviewing the source files I found a driver for a ST7565 display, the ports of KK2.1's ATmega644 the display is connected to and the ports where the keys of a KK2.1 are polled by the ATmega644. Since the KK Programmer seems to work with newer firmwares I reasoned that it must be a dumb device – not more than a breakout box of the display and the buttons normally attached to a KK2.0.

The user connections of the KK2.1HC are labeled on the case – unfortunately the programming headers are not. After opening up the KK2.1HC I found the two headers JP1 and JP2. JP1 is the ISP connection to upgrade the firmware, JP2 the connection of the KK Programmer. Using my multimeter I managed to figure out which pin of JP2 is connected to which port of the ATmega644.



JP1 ISP (orange marks)

Pin	Function	Pin	Function
1	MISO	4	VCC
2	SCK	5	MOSI
3	RST	6	GND

JP2 Programmer (red marks)

Pin	Port	Function	Pin	Port	Function
1	VCC	VCC	7	PD5	CSL
2	PB7	Key 1 / Left/ Back	8	PD6	RESET
3	PB6	Key 2 / Up	9	PD7	A0
4	PB5	Key 3 / Down	10	PD4	SCLK
5	PB4	Key 4 / Right / Menu	11	PD1	MOSI
6	GND	GND	12	?	?

JP2 has thinner pins and a pitch of 2.0mm. Servo leads or jumper (DuPont) wires for regular 2.54mm pitch don't work. Fortunately I managed to find a 5-pin and a 4-pin cable with a pitch of 2.0mm to connect the display and key functions.

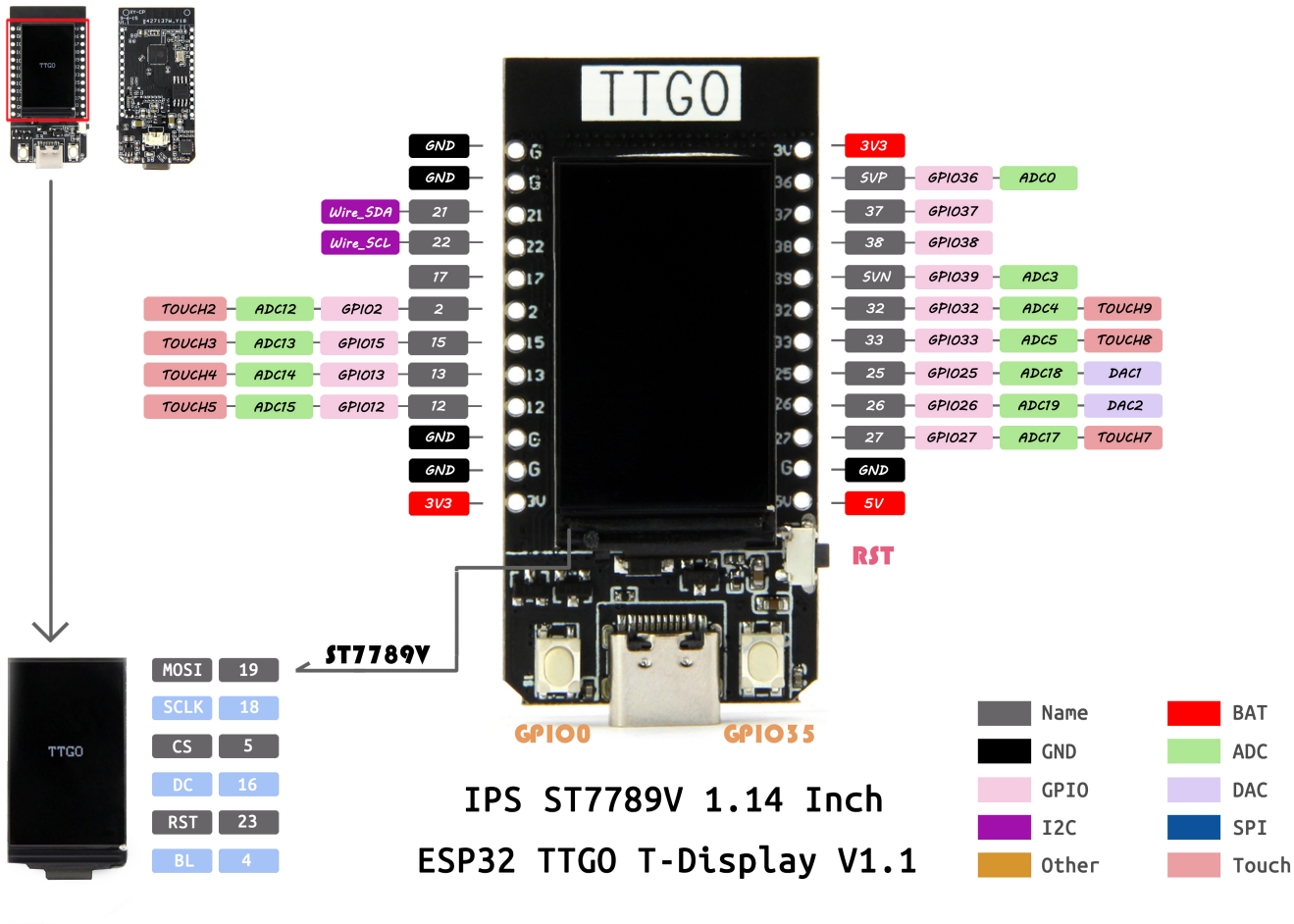
Devices I found in my hobby box

The emulation of the KK Programmer is done with a LilyGO TTGO T-Display (Affiliate Link to Banggood). It is an ESP32 development board with an attached IPS display of resolution 135x240. It works with inputs up to 3.6V, the KK2.1HC uses 5V. So you need a level shifter/converter for all inputs from the flight controller to the ESP32. The easiest way is to build a simple voltage divider of two resistors with e.g. 200Ω and 300Ω. I use resistors with values of 220Ω and 240Ω. That results in logical high levels of roughly 2.5V which is enough for the ESP32.

The manual of the ST7565 display shows that data is transferred over SPI in mode 3 (CPOL 1, CPHA 1). The TTGO has one free hardware SPI, it suffices to connect CSL, SCLK and MOSI. The display's protocol needs an additional connection to input A0 so in total you have to build 4 voltage dividers for 4 signals between the KK2.1HC and the TTGO.

Due to the lack of 4 push buttons I use a 4-way joystick on a HW-504 module. Deflections of the stick will be converted to logical low and high levels the firmware of the KK2.1HC expects to see. It's a natural feeling when navigating through the menu and options of the firmware.

Putting all together



I have chosen the following TTGO pins to connect the JP 2 of the KK2.1HC:

Pin TTGO	Pin JP2	Function	Level shift
GND	GND	GND	no
5V	5V	VCC	no
27	11	MOSI	yes
26	10	SCKL	yes
25	9	A0	yes
33	7	CSL	yes
12	2	Key 1	no
13	3	Key 2	no
15	4	Key 3	no
2	5	Key 4	no

The 4-way joystick HW-504 uses these pins of the TTGO:

Pin TTGO	Pin HW-504
GND	GND
3V	+5V
37	VRx
38	VRy

When holding the pins of the HW-504 upwards, my firmware emits the correct key commands to the KK2.1HC.

If you simply want to upload my firmware "as is" you can use the ESP32 flash download tool (only Windows).

If you're choosing other pins on your dev board please alter the definitions inside of src/main.cpp and recompile. I'm providing my firmware as PlatformIO project inside of Visual Studio Code. But since it's using the Arduino framework the Arduino IDE will work as well:

1. Copy all files under `src` to your sketch folder.
2. Rename `main.cpp` to your `sketch.ino`.
3. Install the TFT_eSPI library.
4. Install the ESP32 board support, the TTGO works as `esp32dev`.

Whether you're using PlatformIO or the Arduino IDE you have to choose the right display configuration with `User_Setup_Select.h` inside of the `TFT_eSPI` lib. Enable the inclusion of `Setup25_TTGO_T_Display.h` for the TTGO T-Display.

For your convenience: Flashing the KK2.1HC

Since the KK2.1HC is delivered with an old 1.6 firmware release I wanted to upgrade to the newest Steveis firmware. I'm using an Arduino Nano as ISP programmer. With the help of Oscar Liang's great blog I connected:

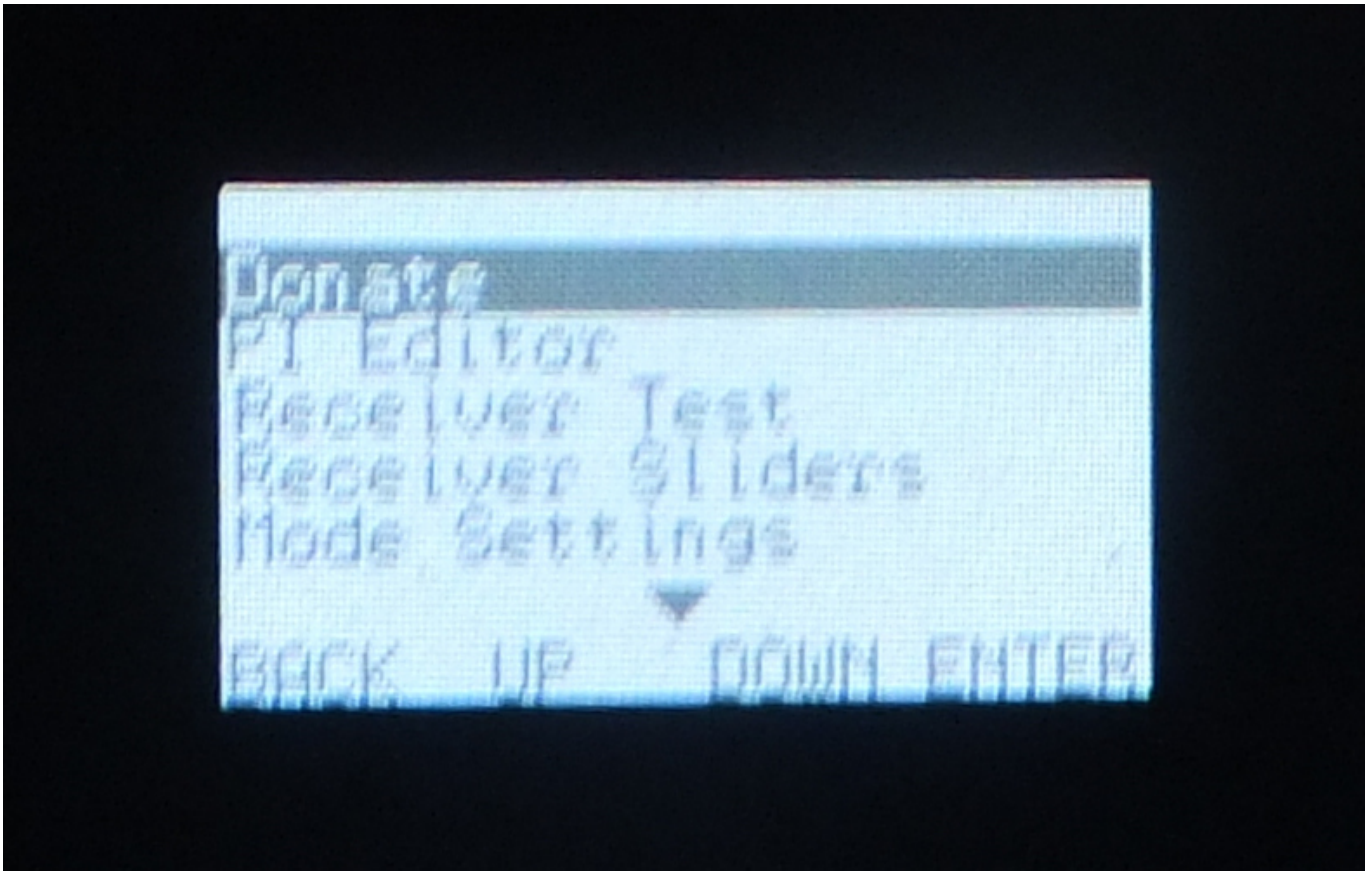
Pin Nano	Pin JP1	Function
GND	6	GND
10	3	RST
11	5	MOSI
12	1	MISO
13	2	SCK

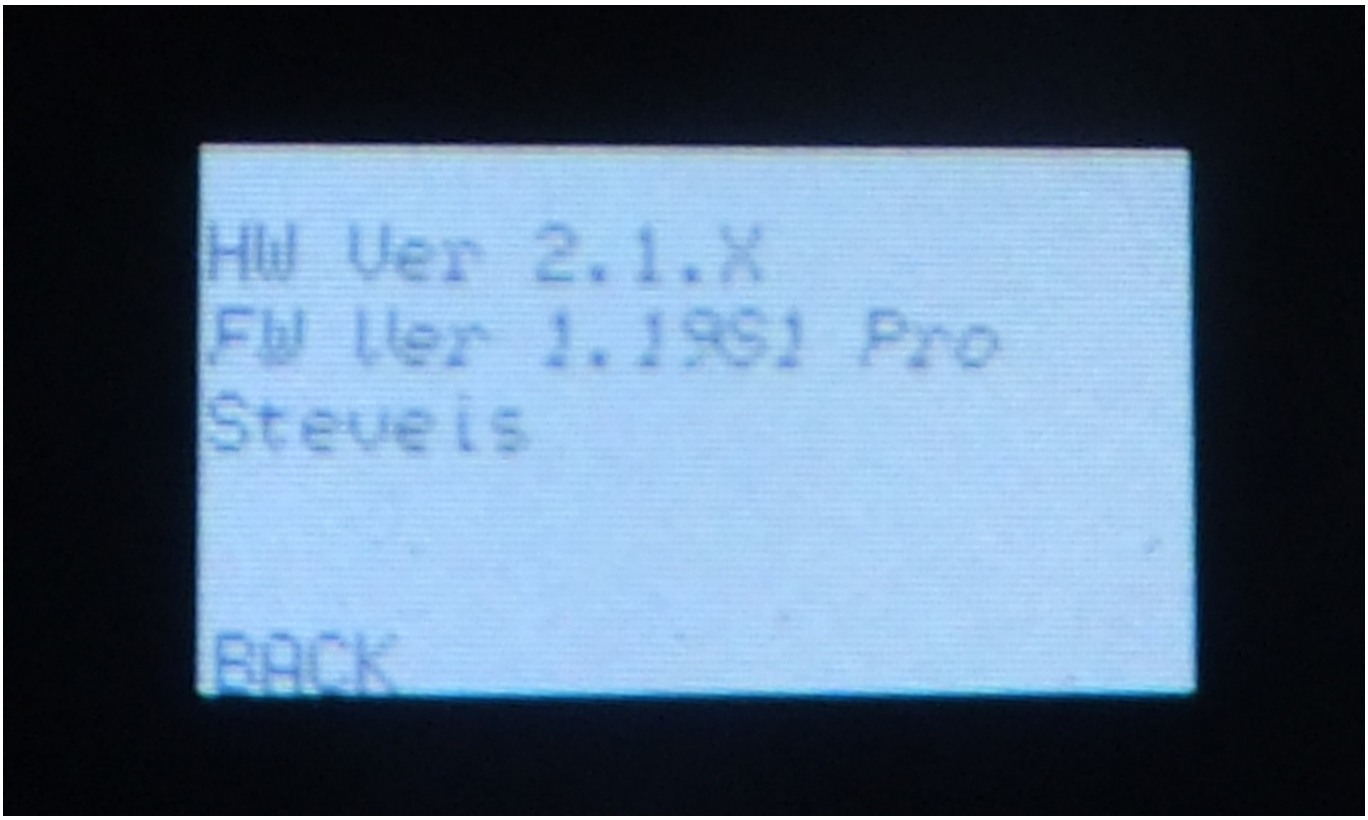
The KK2.1HC should be powered with an external source like an ESC with BEC on Out 1.

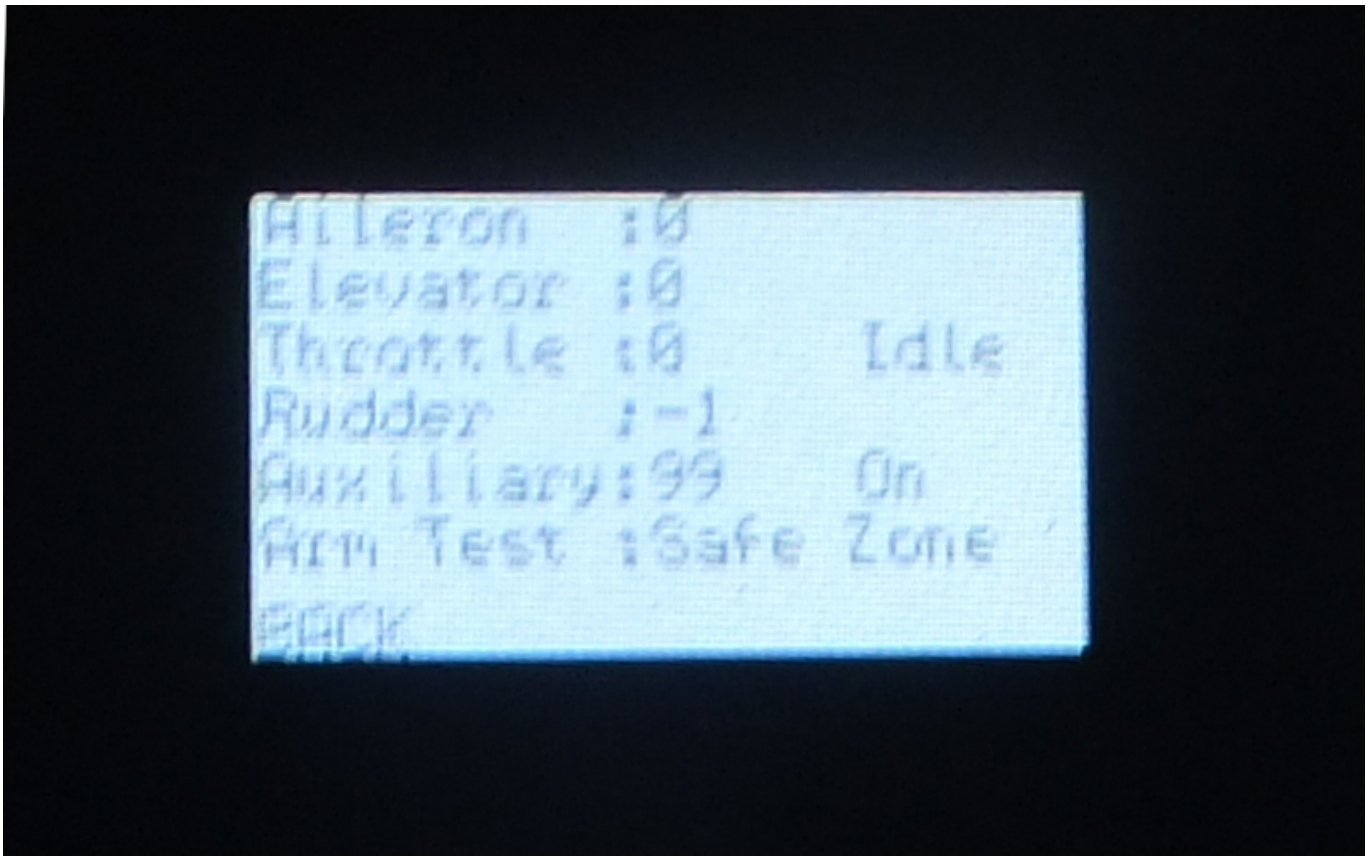
Upload the firmware with `avrdude`:

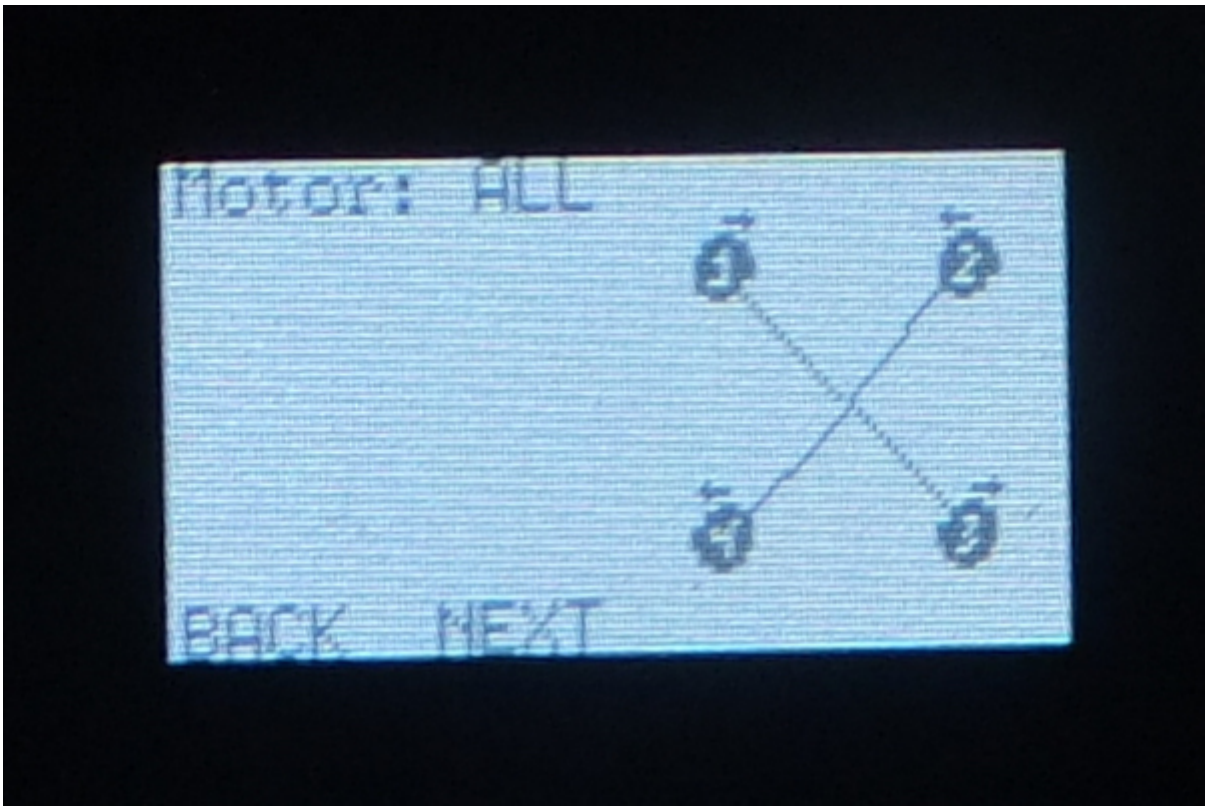
```
avrdude.exe -P COMx -b 19200 -c arduino -p m644p -v -e -U
flash:w:"KK2V1_1V19S1Pro.hex":i
```

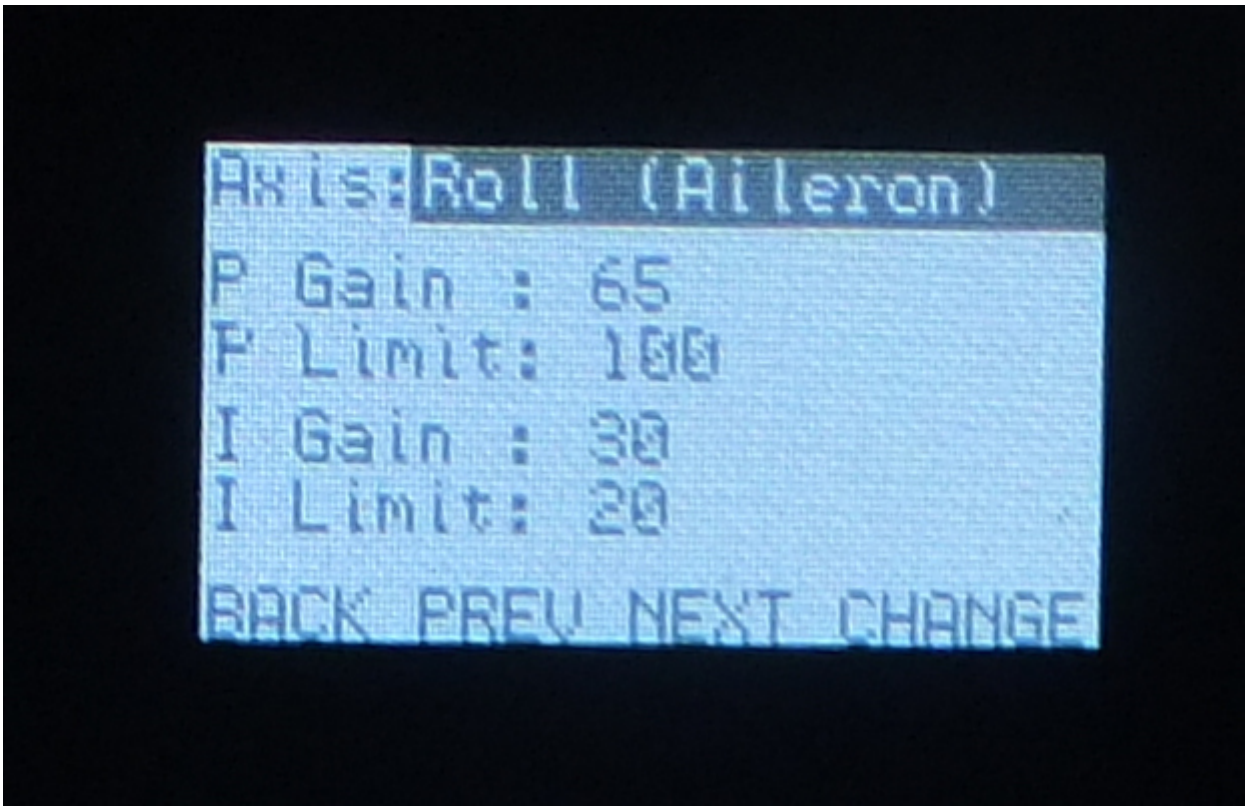
Some screenshots






Aileron :0
Elevator :0
Throttle :0 Idle
Rudder :-1
Auxiliary:99 On
Arm Test :Safe Zone
BACK


Motor: ALL
1
2
3
4
BACK NEXT


Axis:Roll (Aileron)
P Gain : 65
F Limit: 100
I Gain : 30
I Limit: 20
BACK PREV NEXT CHANGE