# Programming

* OpenOCD not recognised? Launch VScode through vscode.bat in wacky racers.
* Did you run openOCD?
* Maybe try run bootloader.

# Board operation

* Board not working like it should? Check pins in the configuration file target.h are correct.
* Make sure you use the correct target (hat vs racer)
* Make sure the correct silk screen label is used. Eg Level shifters silk screens underneath on hat.
* Never copy code! Be especially careful if copying C++ code to C as functions with the same name can do different things.
* Make sure you bracket multi line if statements.
* Try revert back to original micheal code (worst case scenario)

## Power

* ST-Link does not provide power to anything other than the chip
* USB does not provide LED Tape power

## Strange things that are normal

* Unconfigured LED’s will glow dim

# Nothings working

* Did you put it on metal and short pins?
* Power supply is on?
* If you unplugged the board from power, try pressing nrst. Especially if you are using USB

# Future thing to do

* Don’t send ascii characters for radio, send raw bytes which need to be decoded instead, much less data to send.

# Debugger

GDB offers a big list of commands, however the following commands are the ones used most frequently:

* **b main** - Puts a breakpoint at the beginning of the program
* **b** - Puts a breakpoint at the current line
* **b N** - Puts a breakpoint at line N
* **b +N** - Puts a breakpoint N lines down from the current line
* **b fn** - Puts a breakpoint at the beginning of function "fn"
* **d N** - Deletes breakpoint number N
* **info break** - list breakpoints
* **r** - Runs the program until a breakpoint or error
* **c** - Continues running the program until the next breakpoint or error
* **f** - Runs until the current function is finished
* **s** - Runs the next line of the program
* **s N** - Runs the next N lines of the program
* **n** - Like s, but it does not step into functions
* **u N** - Runs until you get N lines in front of the current line
* **p var** - Prints the current value of the variable "var"
* **bt** - Prints a stack trace
* **u** - Goes up a level in the stack
* **d** - Goes down a level in the stack
* **q** - Quits gdb