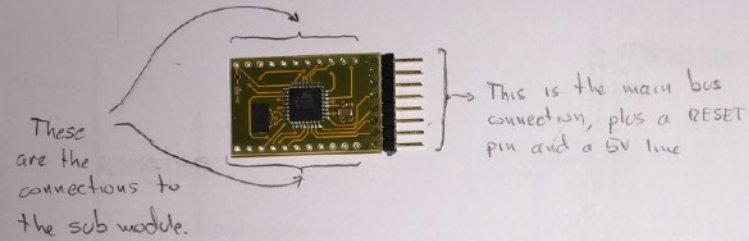


Project documents, part 1.

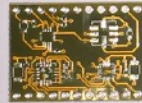
This is a project module:



These are the connections to the sub module.

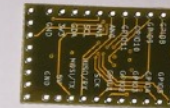
This is the main bus connection, plus a RESET pin and a 5V line

A submodule is a small PCB dedicated to a specific function, like this one:



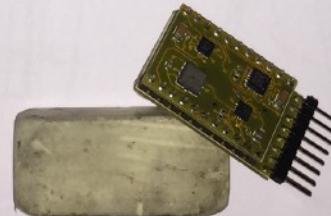
Which was designed to hold four sensors.

This:



is the back side of a module. It shows the standard pin mapping that was adopted to connect to the sub modules.

Here you can see a submodule connected to a module:



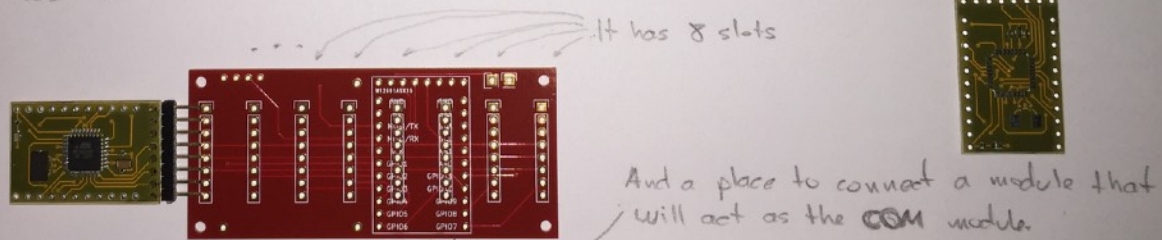
2

Modules are abstract microcontroller boards with a common communication interface and common code to handle it.

Modules can be put to work together to solve a problem. Each module can act independently of the others, or not.

In this system's case, there is a master module. It is called COM, for communications. He is the system courier, calling modules that act as data sources and asking for data, then delivering this data to other decision taking modules, or to execution modules like the servo control.

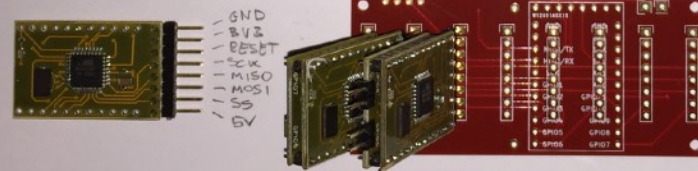
To connect all the system modules I use a pcb that was called the Backbone:



For the unmanned vehicle control system I designed these modules, and sub modules. ③

- ① The MCU module, which contains an ATmega328P microcontroller, a blue led and a few other components.
- ② A nine degrees of freedom sub module, with an accelerometer, a gyroscope, a barometer and a 3D compass.
- ③ A GPS module, which is nothing but a MCU module connected to a GPS.
- ④ A servo control sub module, which needs a MCU module to operate.
- ⑤ A Flight control module, again a MCU module programmed to control the flight.
- ⑥ Two power supply submodules, one to feed the servos with 5V and the other to feed the logic with 3.3V. They too need a MCU module to operate.

So we have:



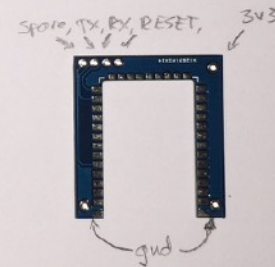
The COM module needs to be connected to the back bone because it uses the main bus to transfer info, and the GPIOs to drive the SS of the other system modules. Remember it is the SPI master of the bus.

④

The backbone PCB also has space to hold and connect to a wifi commercial module via UART.



Here you can see the RN131G module soldered to a support board and placed over the backbone PCB.

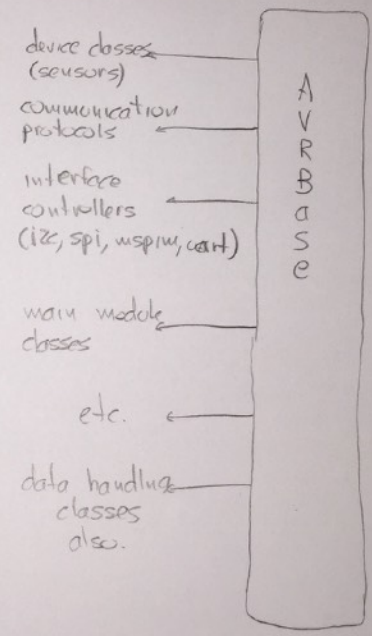


This is the RN131G support board I made. It has only 5 connections to the backbone

⑤

Each MCU module is programmed with code written in C++. The code is specific for the function that the module has in the system. The module codes are divided in classes, all derived from a base class called AVRBase.

AVRBase includes relevant AVR headers and very basic common code that is needed by all its inherited classes. It also includes pin name definitions and other important stuff.



Every module code follows the following diagram:

6

