

Project Specification: Drinkbot
Custom Hardware: Drinkbot (OSHW)
<https://github.com/felixekman/Drinkbot>
Schematics available as DrinkbotPCB.pdf

Main Processing Unit: Spark Core
Secondary Processing Unit: ATmega328P

Important information for System:

8 Pumps, connected to pins D0, D1, A0, A1, A4, A5, A6, A7 on Spark Core
Pumps are controlled via MOSFET's (FQP30N06L)

Alcohol Sensor (MQ3) connected to pin A2 on Spark Core
Temperature Sensor (DS18B20) connected to pin D4 on Spark Core

Secondary Processing Unit controlled via Serial from Spark Core. TX/RX Connected on TX/RX pins on Spark Core.

RGB led (Neopixel) connected to Spark Core on pin D7.

Functions and control sequences:

Commands arrives to Spark Core via Curl/Spark.function/Spark.variable

:Command syntax:

For Drink order/Pump order:

Ex: P0:10;P1:00;P2:20;P3:05;P4:07;P5:15;P6:00;P7:10
Ex 2: P0:XX;P1:XX;P2:XX;P3:XX;P4:XX;P5:XX;P6:XX;P7:XX

The command will always have this format P0,P1 etc. corresponds to the pump, the following two values is the cl value that the specific pump should produce. This cl value corresponds to a specific "on time" value for each cl. So for example for Pump 1, P1:10, i.e. 10 cl would equal a delay of $500 \times 10 = 5000$. This is to be set by variable that can be changed via Spark.function call. When command is finished and all pumps are done the API response should return an "All done", if any error occurs during the run this should also be returned. For information on this please refer to:

<http://docs.spark.io/firmware/> and <http://docs.spark.io/api/>

Also, very important! Incoming command must be executed in "parallel", I.e. I want to issue a curl commands that trigger all the pumps to run for a set time. But I want to be able to trigger them all so I don't have to wait for each pump to finish before continuing to the next one, i.e. not a cascading run.

There is some sample code available for this in this file on Github (SampleCode.txt that gives some pointers on how we want it to be done.

For Alcohol Sensor order:

When a command comes in from the Server the Spark Core should turn on the Alcohol Sensor, wait [delay(6000)], read the value from it and convert it to BAC and return it in the response. This is done by simply reading the analogRead(0) value of pin A2 and then calculating % BAC with the analogRead value and the calculations available here: <https://code.google.com/p/ece387breathalyzer/wiki/theCode>

Example:

We issue a Curl order on a Server side app via this command:

```
curl
https://api.spark.io/v1/devices/0123456789abcdef01234567/alcohol?access_token=123412341234123412341234123412
```

```
// EXAMPLE RESPONSE
{
  "cmd": "VarReturn",
  "name": "alcohol",
  "result": 0.06,
  "coreInfo": {
    "last_app": "",
    "last_heard": "2014-08-22T22:33:25.407Z",
    "connected": true,
    "deviceID": "53ff6c065075535119511687"
  }
}
```

Example: On Spark: "Spark.variable("alcohol", &alcohol, INT);"

On Spark Core's side it simply turns on the alcohol sensor, waits a set amount of time, then reads the value from the alcohol sensors analog voltage and response and calculates the estimated BAC.

For Temperature Sensor order:

Same as Alcohol Sensor order, but instead reads value from DS18B20 sensor and outputs it to a temperature variable instead. The library for the DS18B20 is available in the Spark IDE (SPARK-DALLAS-TEMPERATURE) Sample code can be found here:

<http://www.reuk.co.uk/wordpress/ds18b20-temperature-measurement-with-spark-core/>