

Agenda

See Midterm Updates

Announcement

Serial Intro

Serial Workshop



Ceci n'est pas une pipe.

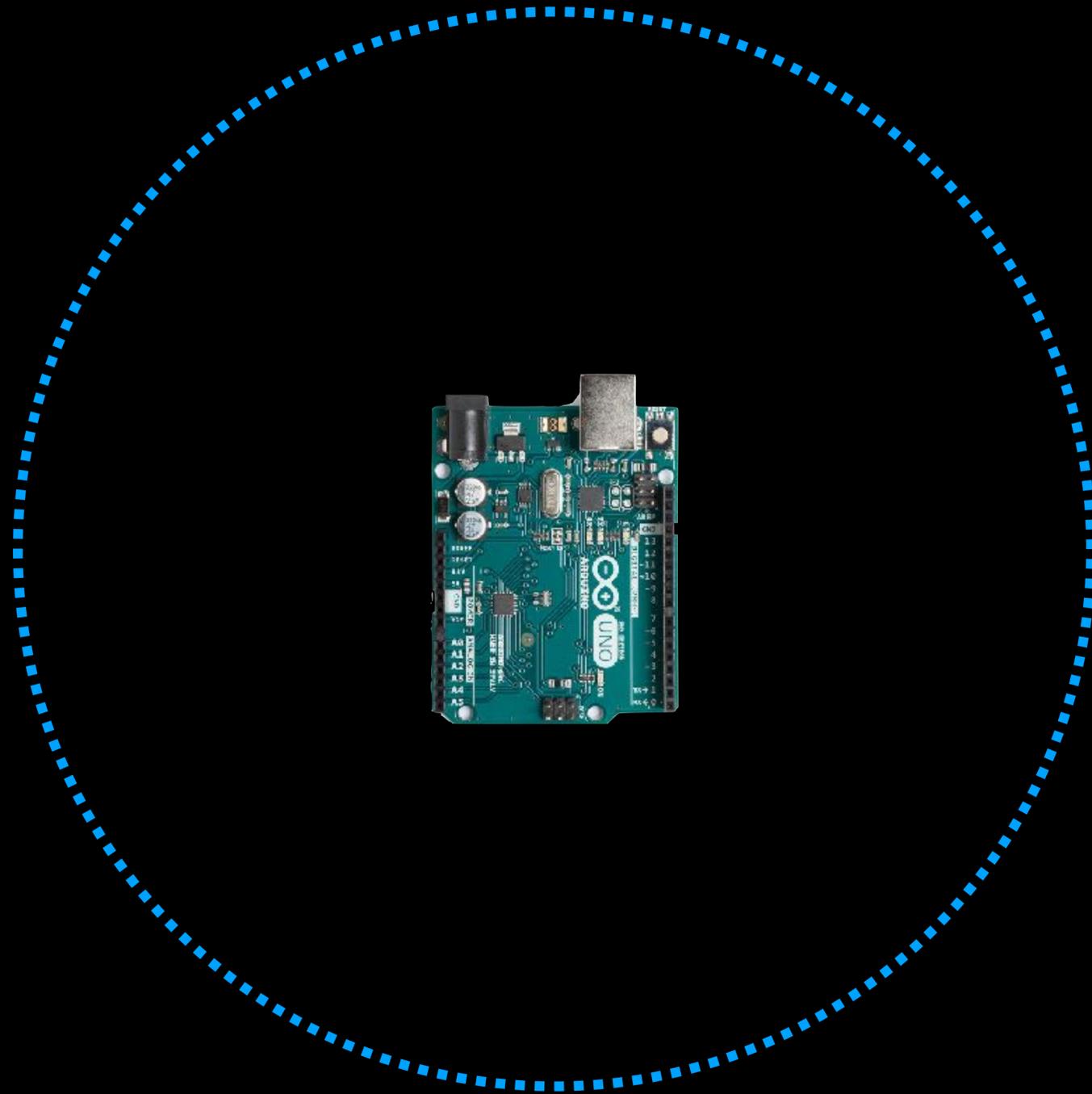
Magritte

This is not a painting by Magritte

This is not the idea that this is not a painting by Magritte...

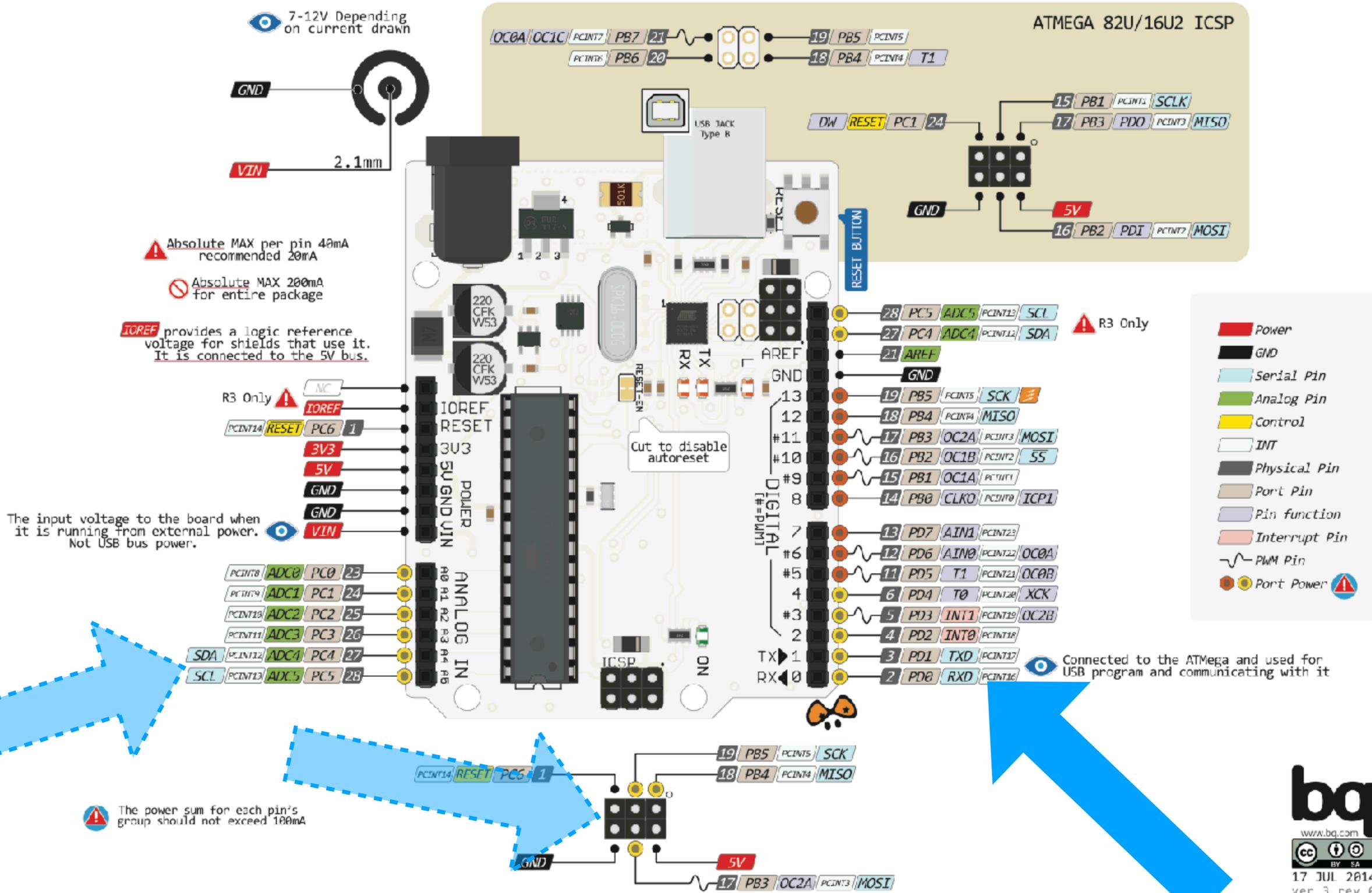
Why Serial?





Arduino is lonely

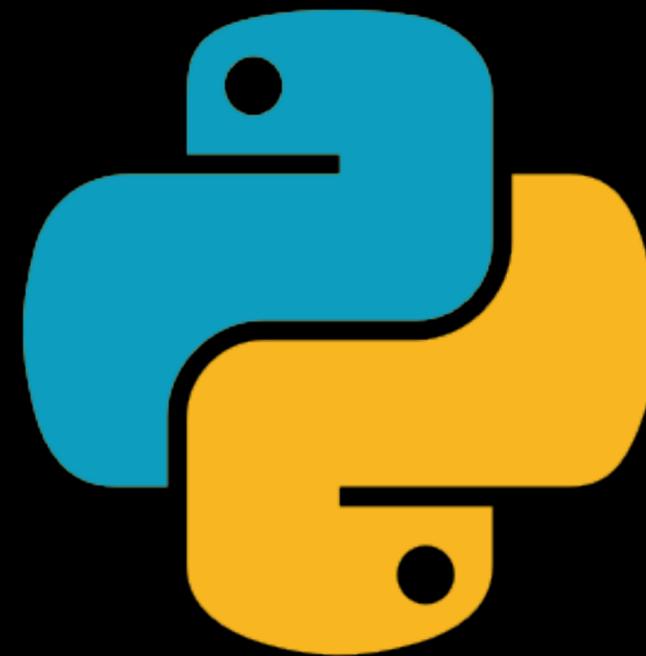
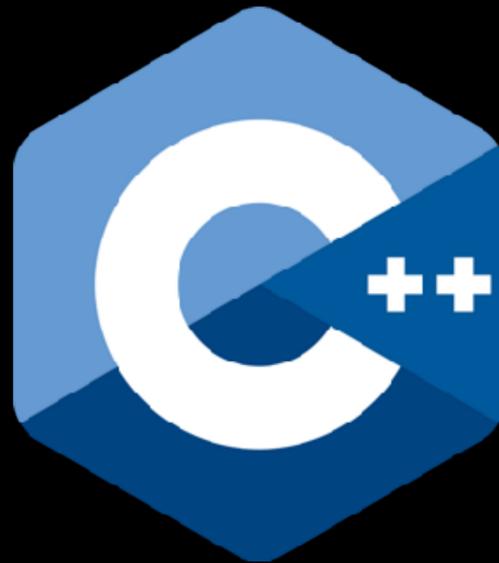
UNO PINOUT



Arduino can talk to the world three ways

PO5 *JS

p5.js



Fingerprint Sensors

“This great GT-521F52 fingerprint module from ADH-Tech communicates over TTL Serial so you can easily embed it into your next project.” - Sparkfun product page



Adafruit's

GSM + GPS

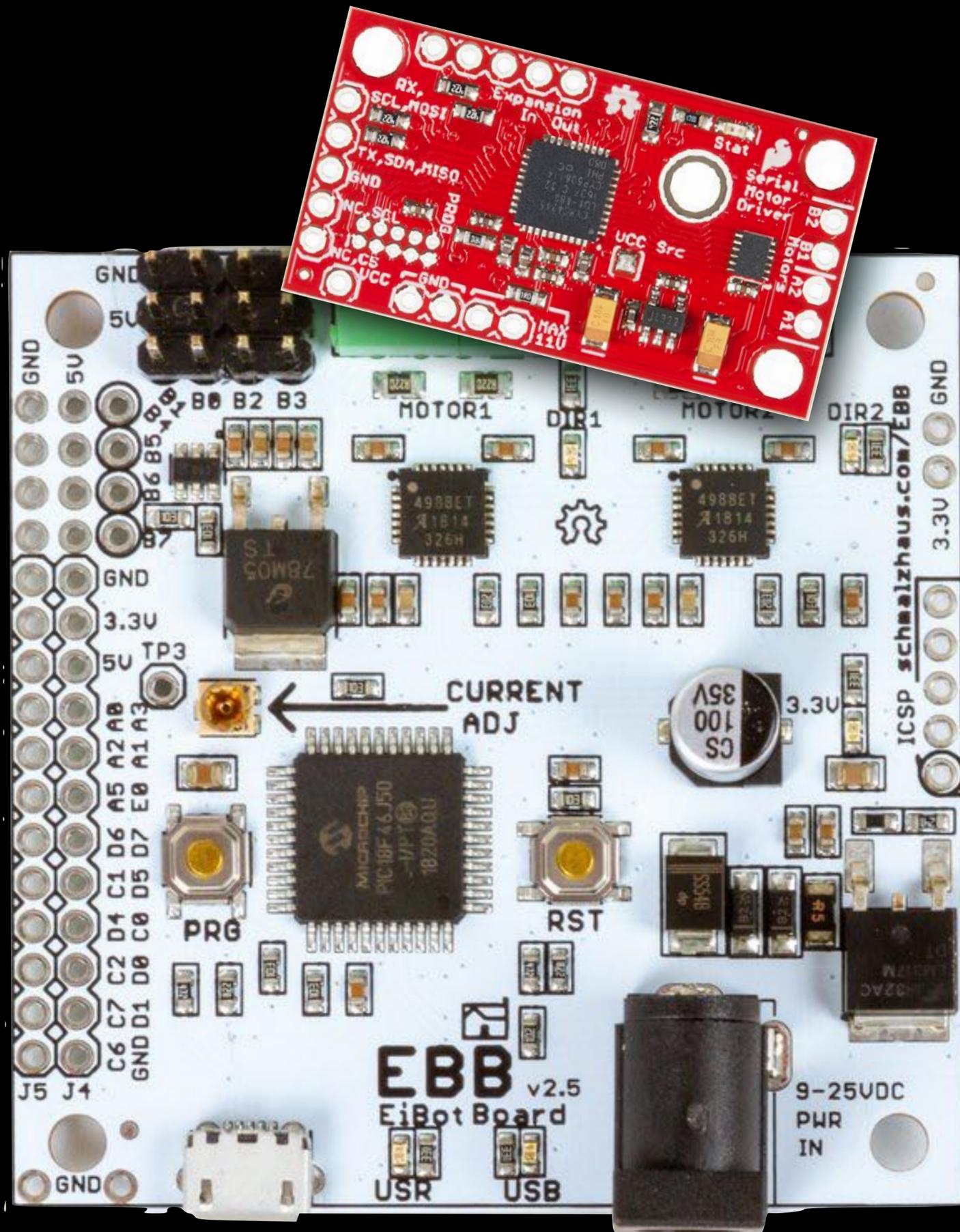
“On its own, this module can't do anything. It requires a microcontroller to drive it! We suggest and use an Arduino but **any 3-5V microcontroller with a UART can send and receive commands over the RX/TX pins.**”

- Adafruit product page

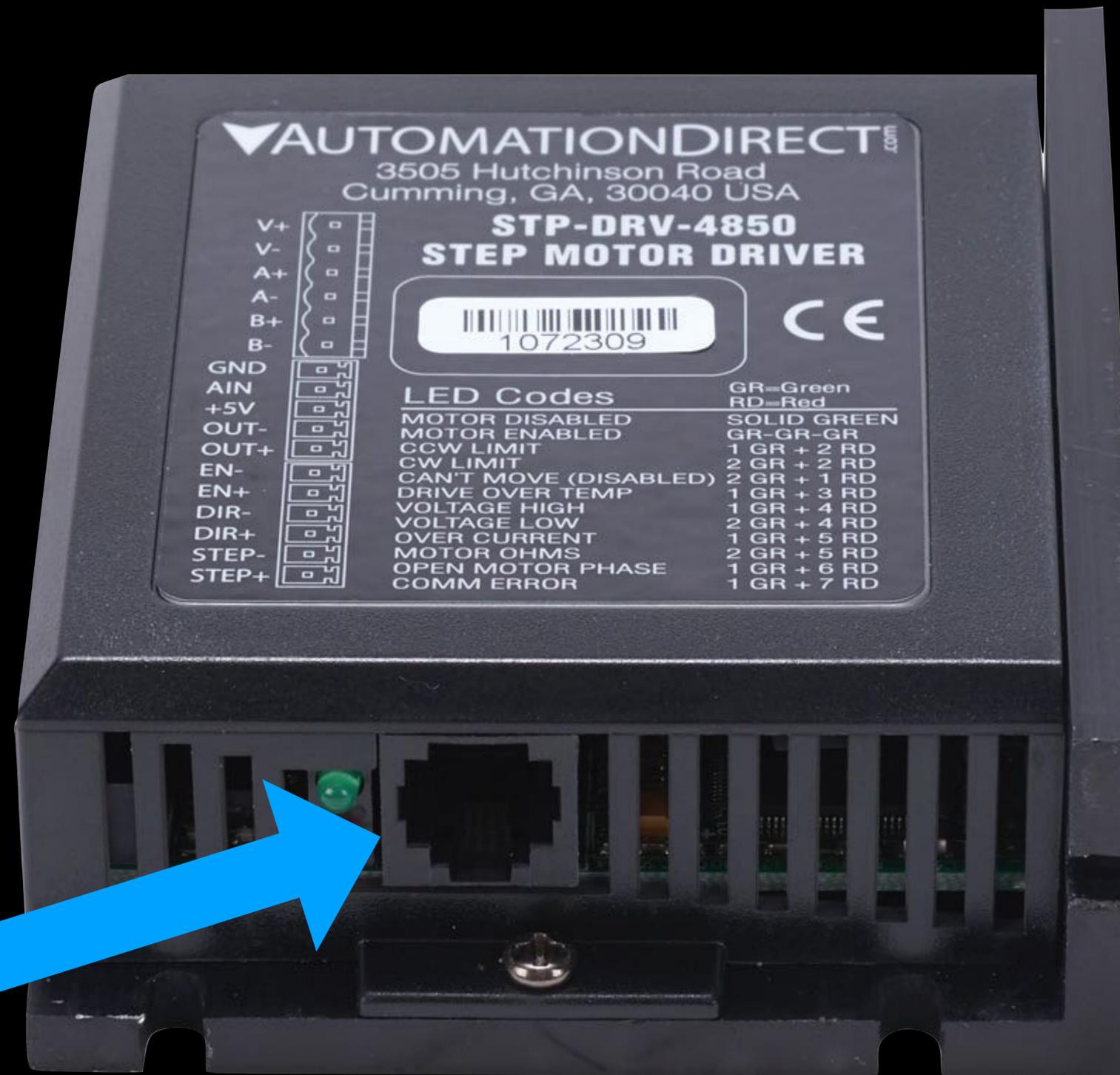


Cheap Motor Control

“You can easily **type commands** to it using a **terminal emulator**, or write your own application to send commands for moving the stepper motors.” - EEB Product page



Mid-priced Motor Control



AUTOMATIONDIRECT
STP-DRV-4850
STEP MOTOR DRIVER



CE
RoHS

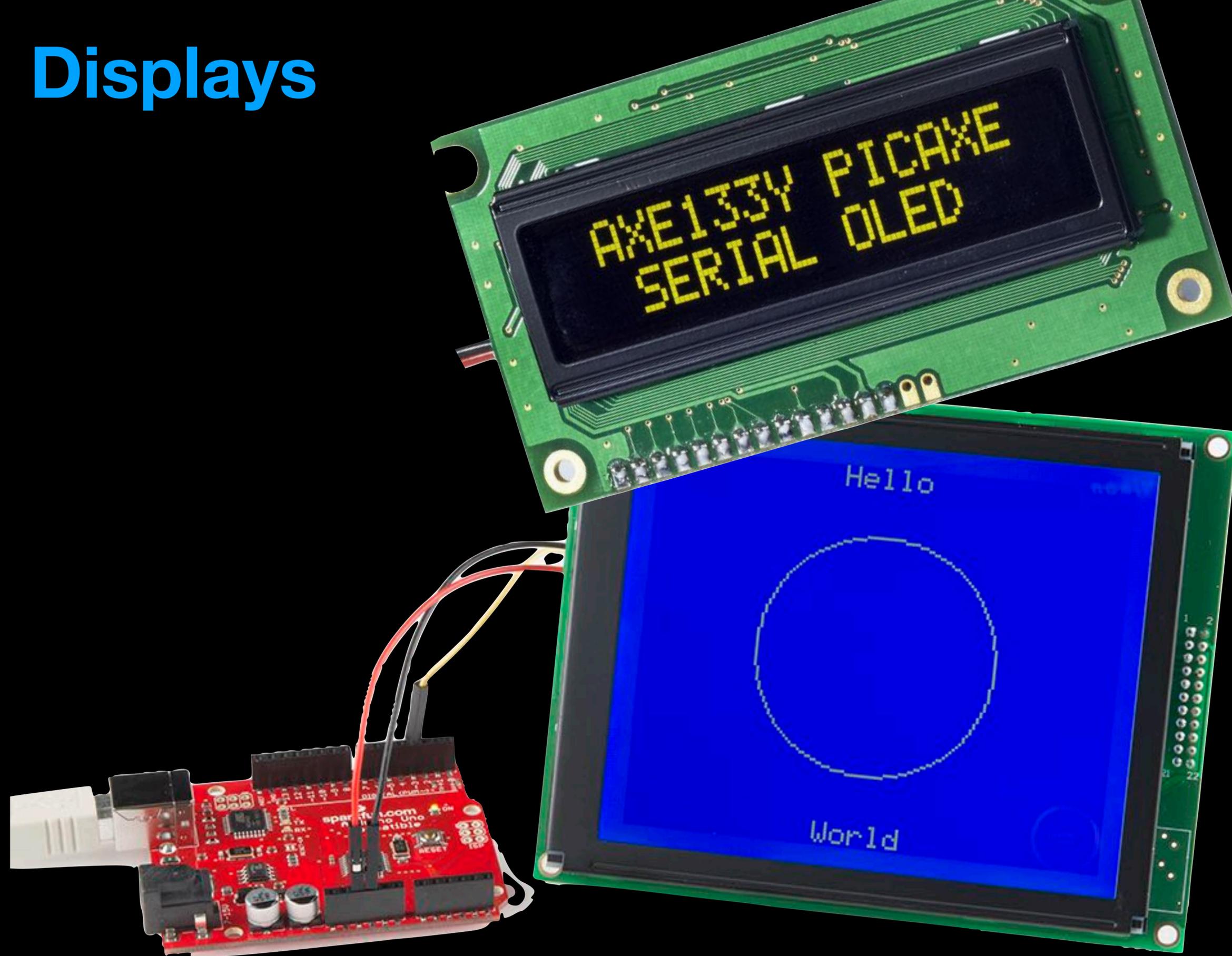
V+
V-
A+
A-
B+
B-
GND
AIN
+5V
OUT-
OUT+
EN-
EN+
DIR-
DIR+
STEP-
STEP+

LED Codes

MOTOR DISABLED	GR=Green
MOTOR ENABLED	RD=Red
CCW LIMIT	SOLID GREEN
CW LIMIT	1 GR + 2 RD
CAN'T MOVE (DISABLED)	2 GR + 2 RD
DRIVE OVER TEMP	1 GR + 1 RD
VOLTAGE HIGH	2 GR + 3 RD
VOLTAGE LOW	1 GR + 4 RD
OVER CURRENT	2 GR + 4 RD
MOTOR OHMS	1 GR + 4 RD
OPEN MOTOR PHASE	2 GR + 5 RD
COMM ERROR	1 GR + 5 RD
	1 GR + 6 RD
	1 GR + 7 RD

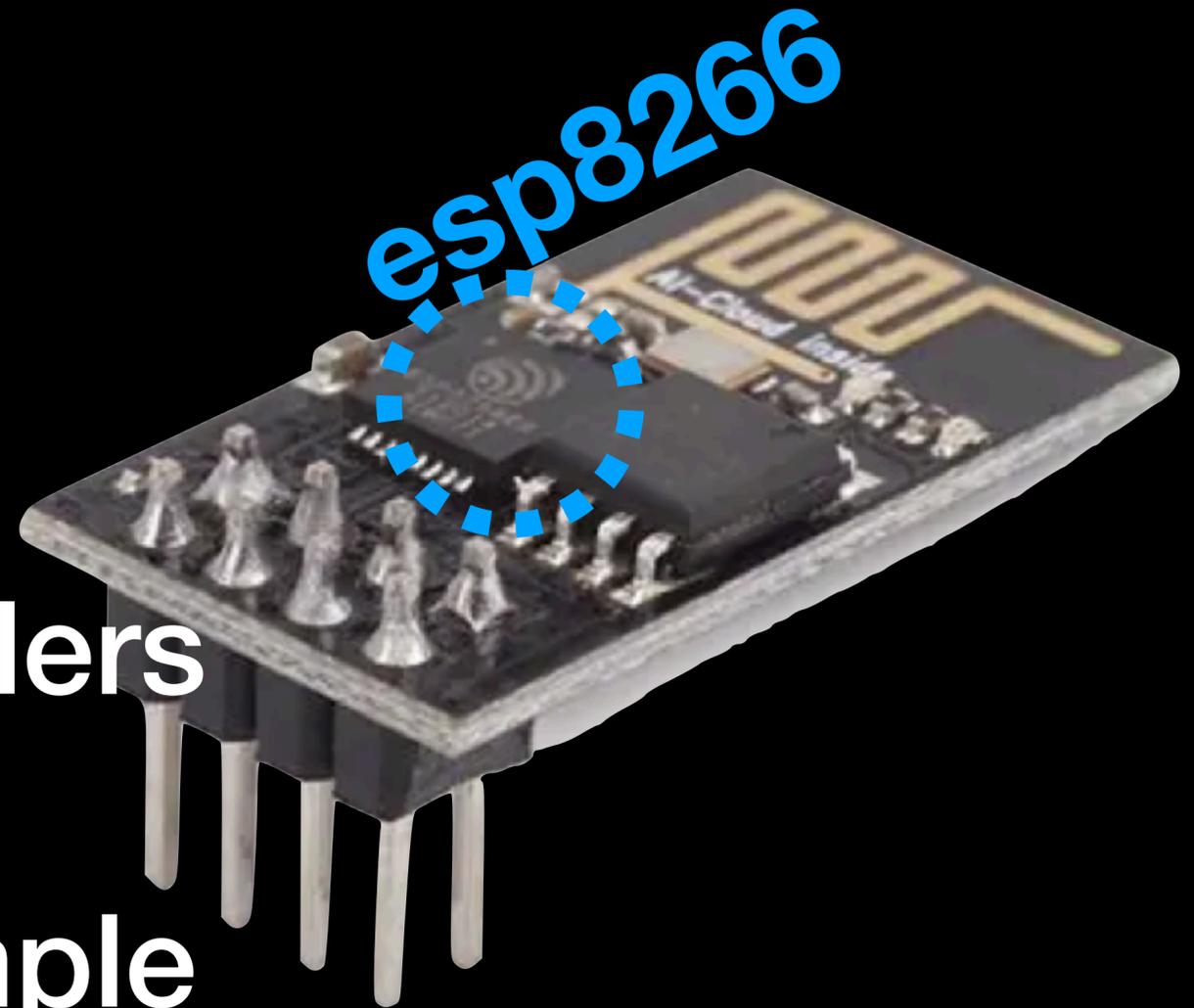


Displays



Wifi-serial circa 2014

“This small module allow[ed] microcontrollers to connect to a Wi-Fi network and make simple TCP/IP connections using **Hayes-style [serial] commands.**”
- Wikipedia

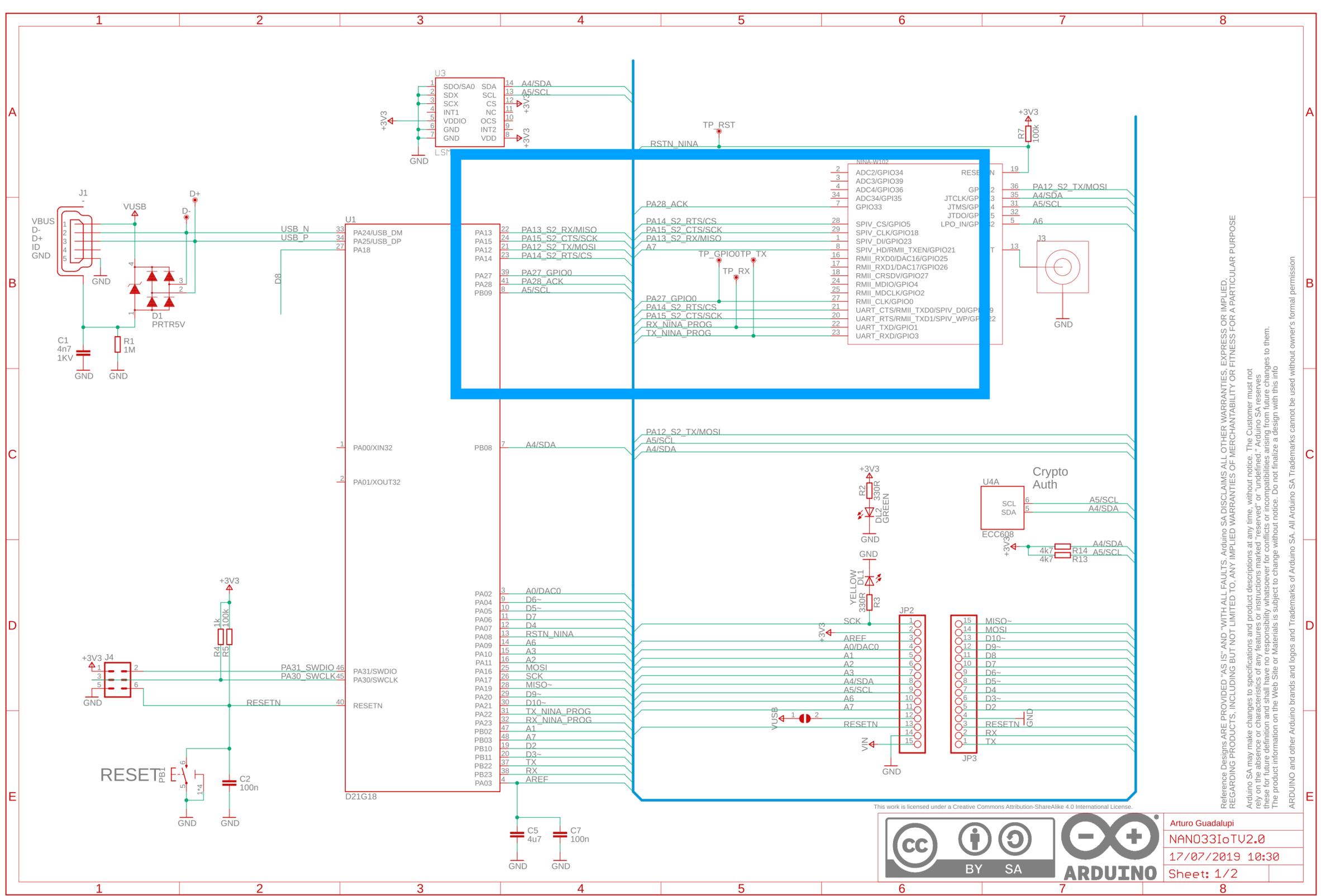


Wifi in 2018

“In late October 2014, Espressif Systems released a software development kit (SDK) for programming the chip directly, which removed the need for a separate microcontroller.”

- Wikipedia



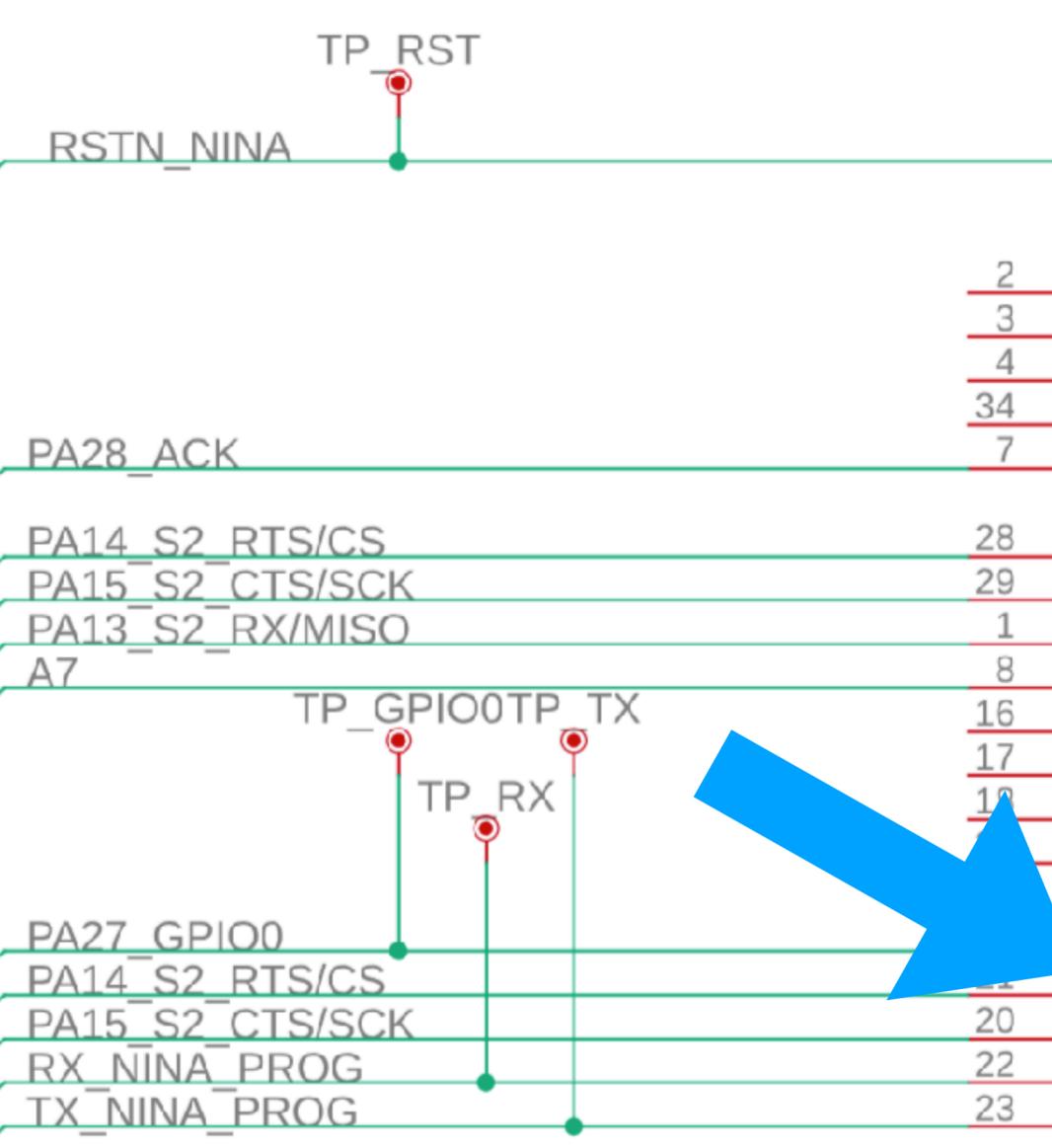
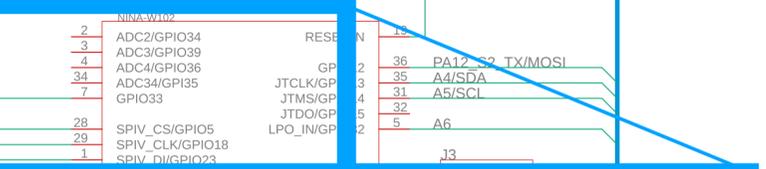
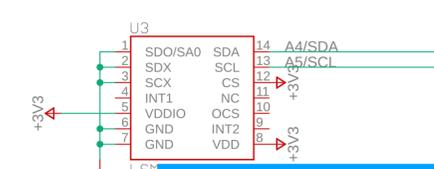
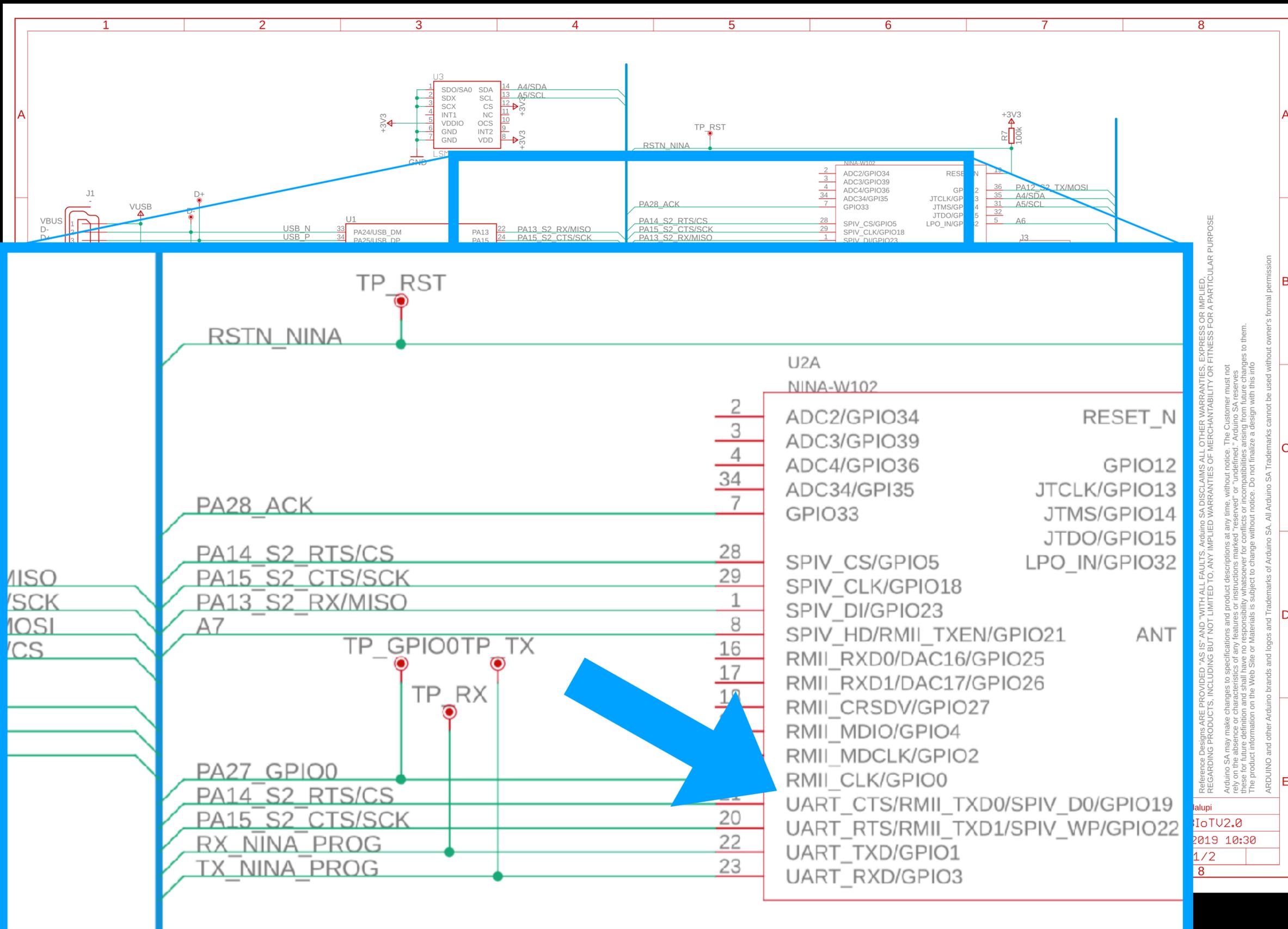


Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. ARDUINO SA DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ARDUINO SA may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Arduino SA reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this info. ARDUINO and other Arduino brands and logos and Trademarks of Arduino SA. All Arduino SA Trademarks cannot be used without owner's formal permission.

This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.



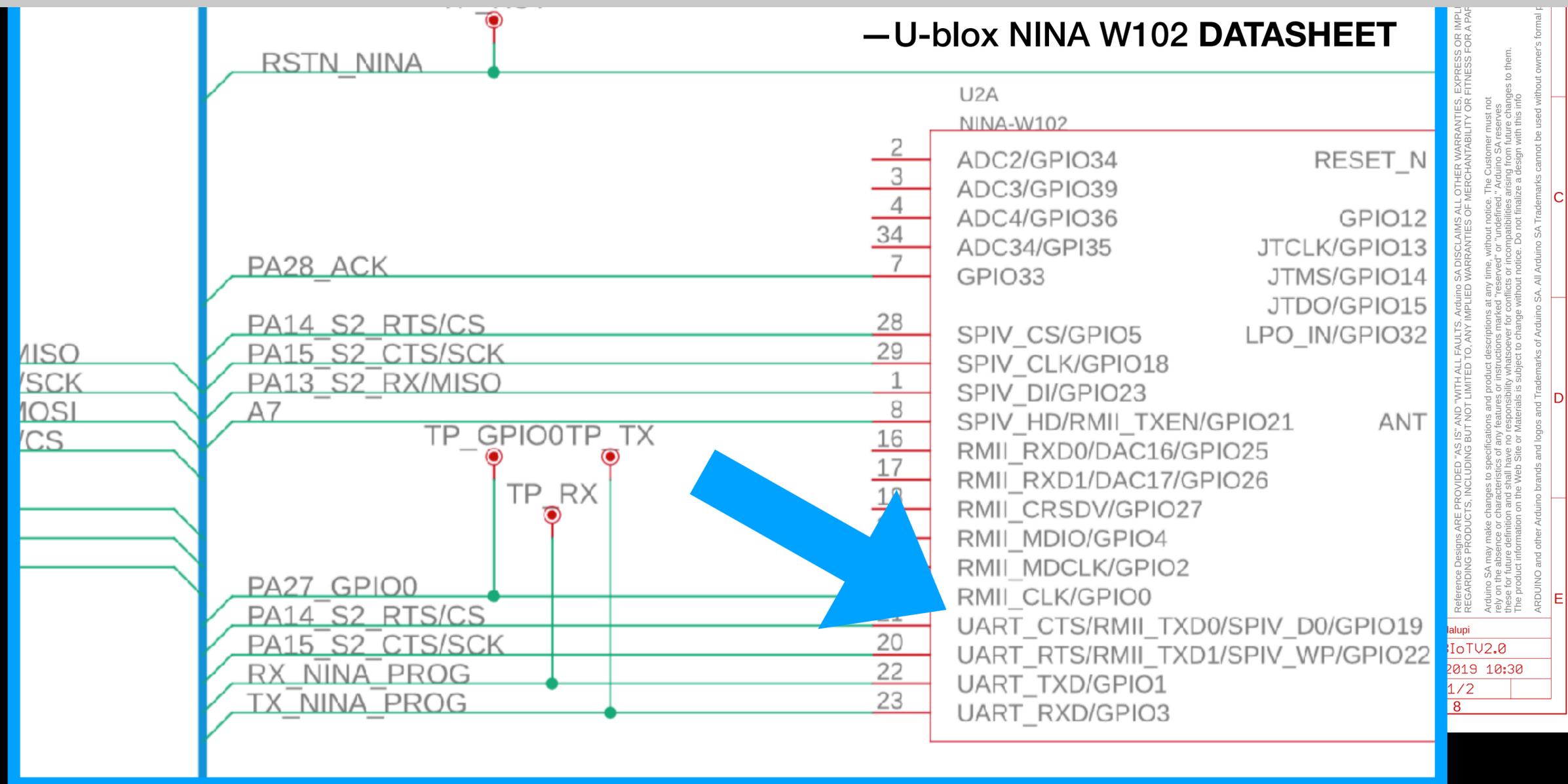
Arturo Guadalupi
 NAN033IoT V2.0
 17/07/2019 10:30
 Sheet: 1/2



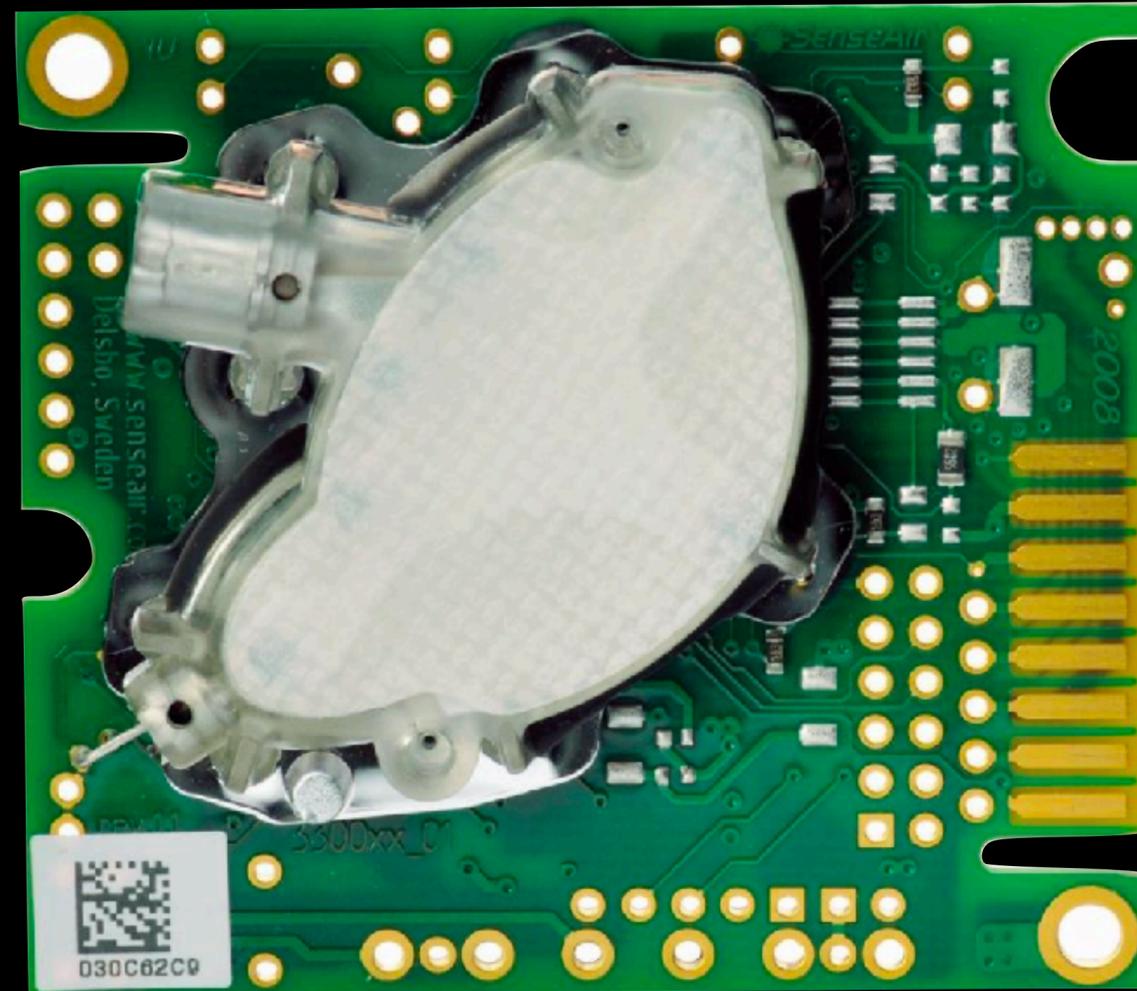
U2A	
NINA-W102	
2	ADC2/GPIO34
3	ADC3/GPIO39
4	ADC4/GPIO36
34	ADC34/GPI35
7	GPIO33
28	SPIV_CS/GPIO5
29	SPIV_CLK/GPIO18
1	SPIV_DI/GPIO23
8	SPIV_HD/RMII_TXEN/GPIO21
16	RMII_RXD0/DAC16/GPIO25
17	RMII_RXD1/DAC17/GPIO26
19	RMII_CRSDV/GPIO27
21	RMII_MDIO/GPIO4
22	RMII_MDCLK/GPIO2
23	RMII_CLK/GPIO0
24	UART_CTS/RMII_TXD0/SPIV_D0/GPIO19
20	UART_RTS/RMII_TXD1/SPIV_WP/GPIO22
22	UART_TXD/GPIO1
23	UART_RXD/GPIO3
	RESET_N
	GPIO12
	JTCLK/GPIO13
	JTMS/GPIO14
	JTDO/GPIO15
	LPO_IN/GPIO32
	ANT

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. ARDUINO SA DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. ARDUINO SA may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". ARDUINO SA reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this info. ARDUINO and other Arduino brands and logos and Trademarks of Arduino SA. All Arduino SA Trademarks cannot be used without owner's formal permission.

2.7.1 UARTs NINA-W10 modules have three UART interfaces, UART0 to UART2. Each interface provides asynchronous communication support for RS232, RS485, and IrDA standards (with external drivers). UART0 serves as the primary interface port. The maximum speed for all UART interfaces is 5 Mbit/s.



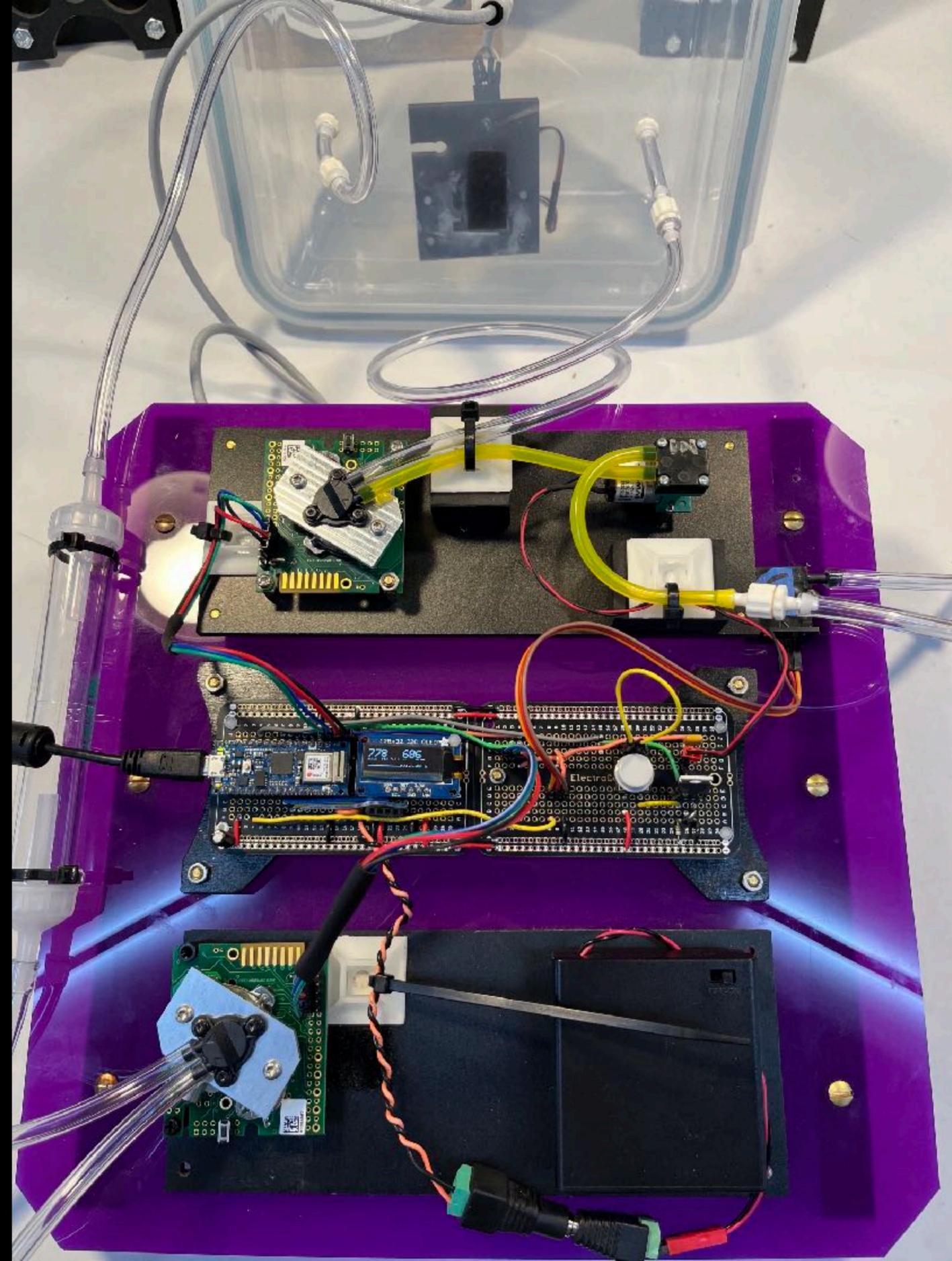
CO2 Sensor



UART Serial, I2C *and*
analog outputs

CO2 Sensor Suite:

2x serial to K30
1x serial to computer
WIFI to cloud (option)
I2C to sensors and display



Fans

Interior electronics port

Gas sampling port

Diffusor

Measurement electronics

Water supply

Airspeed sensor

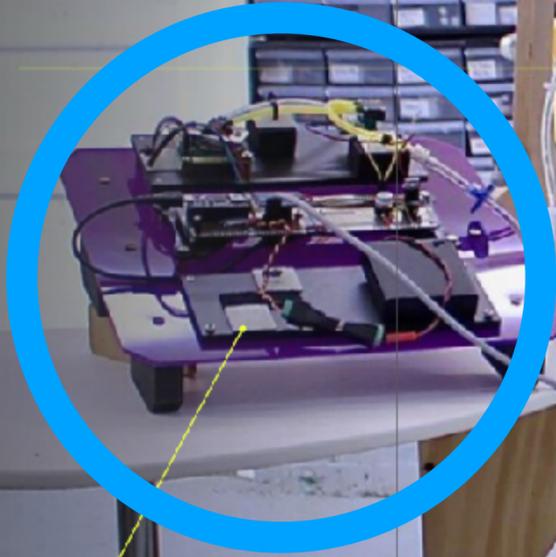
Sample return line

Sprinklers

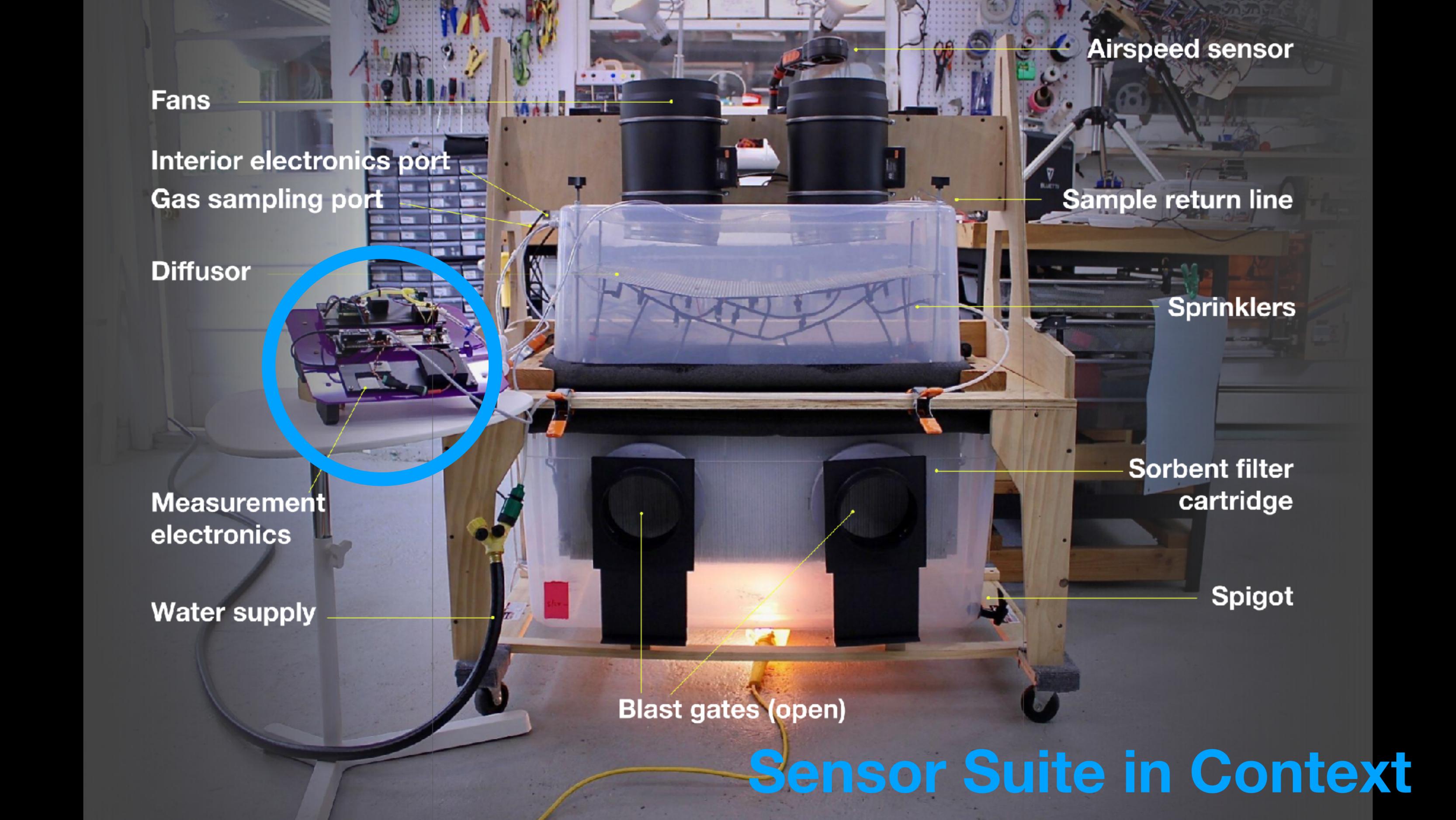
Sorbent filter cartridge

Spigot

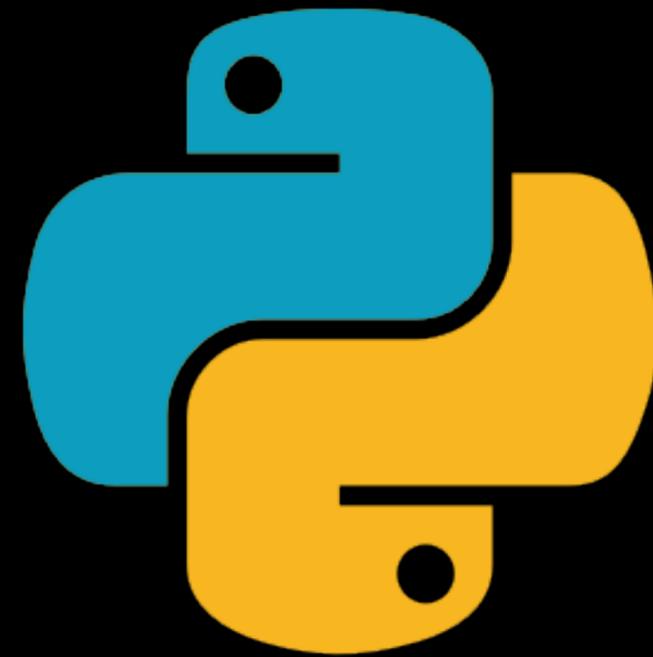
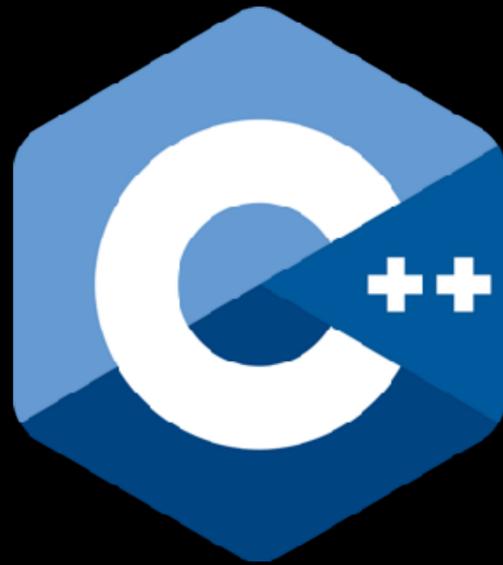
Blast gates (open)

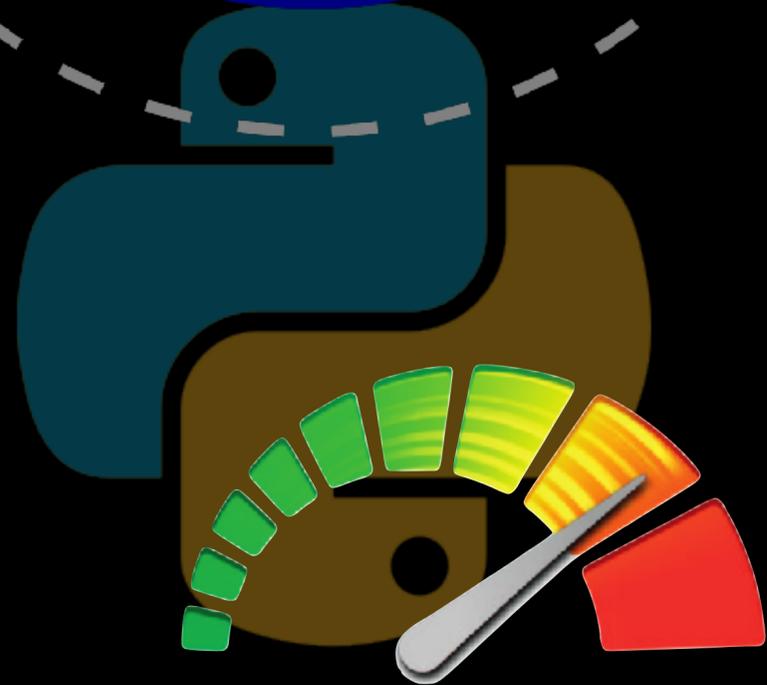


Sensor Suite in Context



p5.js



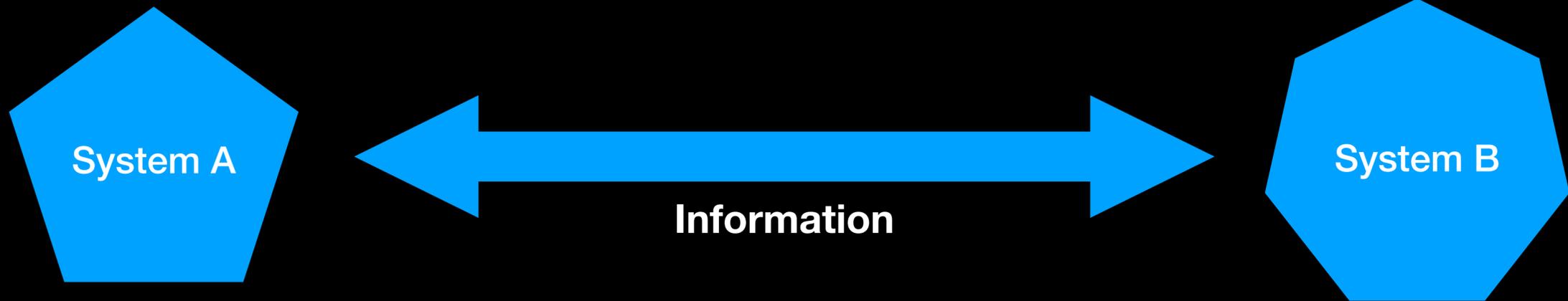


Java

etc...



etc...



Mind A



Mind B



Information

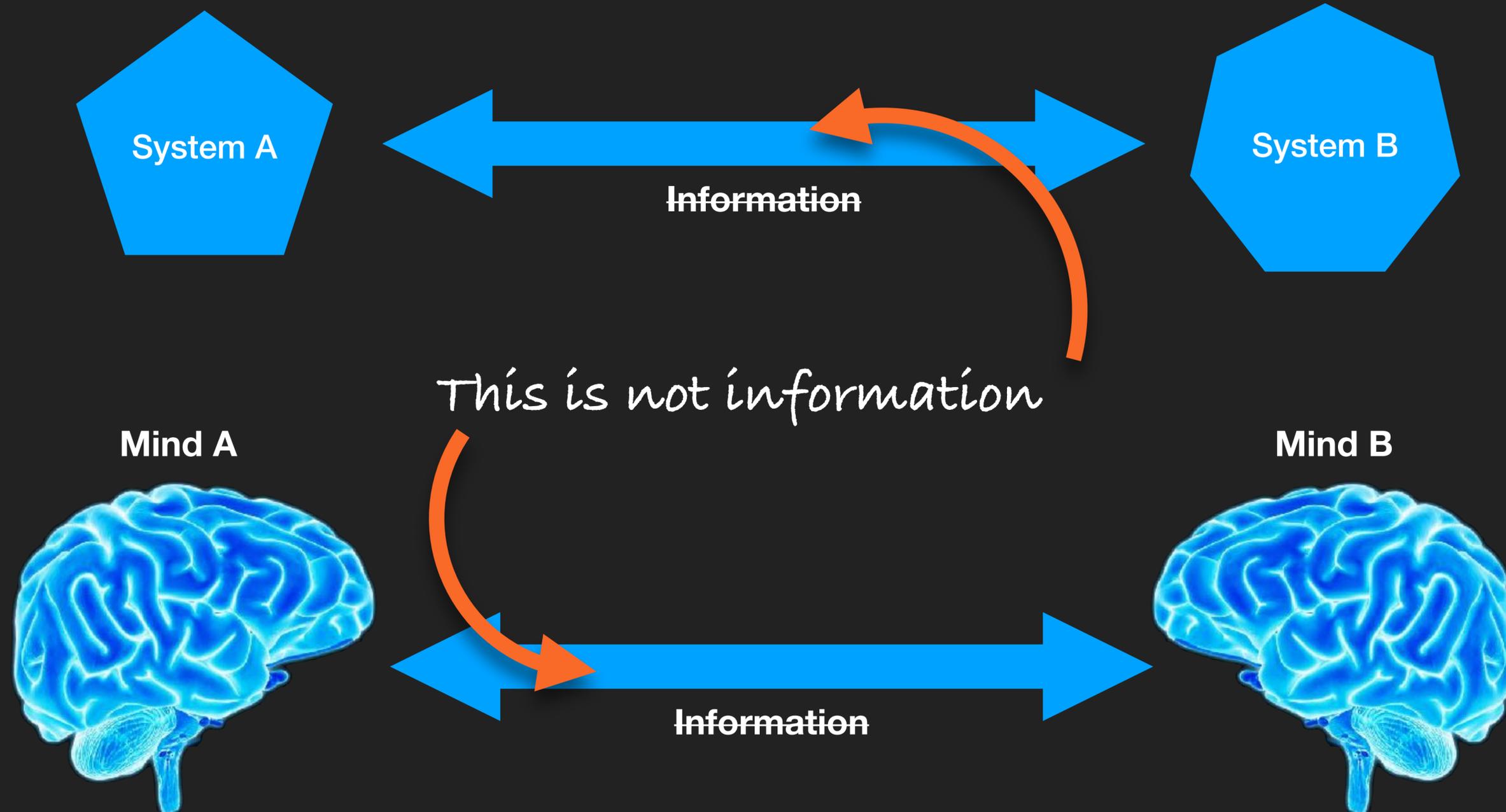


Ceci n'est pas une pipe.

Magritte

This is not a painting by Magritte

This is not the idea that this is not a painting by Magritte...



System A

System B

Information

This is not information

Mind A

Mind B

Information



System A

Voltage changing from moment to moment

System B

Information

Variable declaration in code:

```
byte b = 64;
```

Bits in memory:

0100 0000

Variable name:

"b"

Value to us (decimal):

"64"

ASCII symbol

"@"

Method:

```
Serial.write(b)
```

Bits sent out serially:

0100 0000

64->"@"

```
Serial.print(b)
```

0011 0100

52->"4"

0011 0110

54->"6"

```
Serial.println(b)
```

0000 1101

13->"CR"

0000 1010

10->"LF"

0011 0100

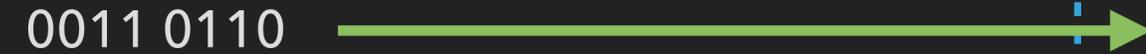
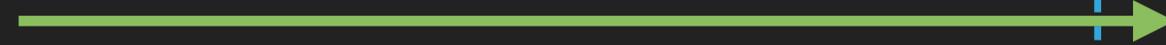
52->"4"

0011 0110

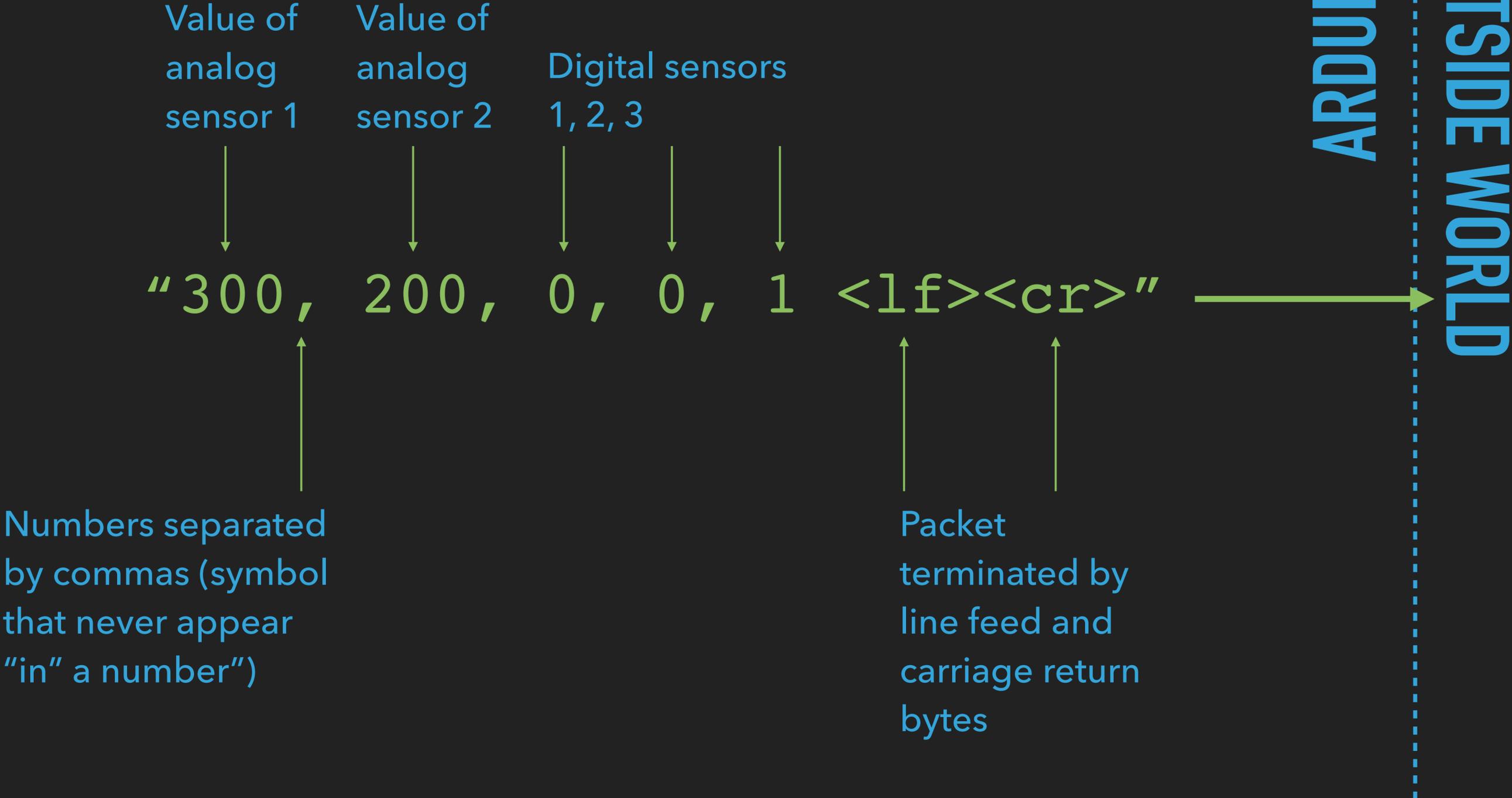
54->"6"

ARDUINO

OUTSIDE WORLD



Sender generates data "packet" as a string



Receiver breaks up packet into parts

OUTSIDE WORLD

P5

