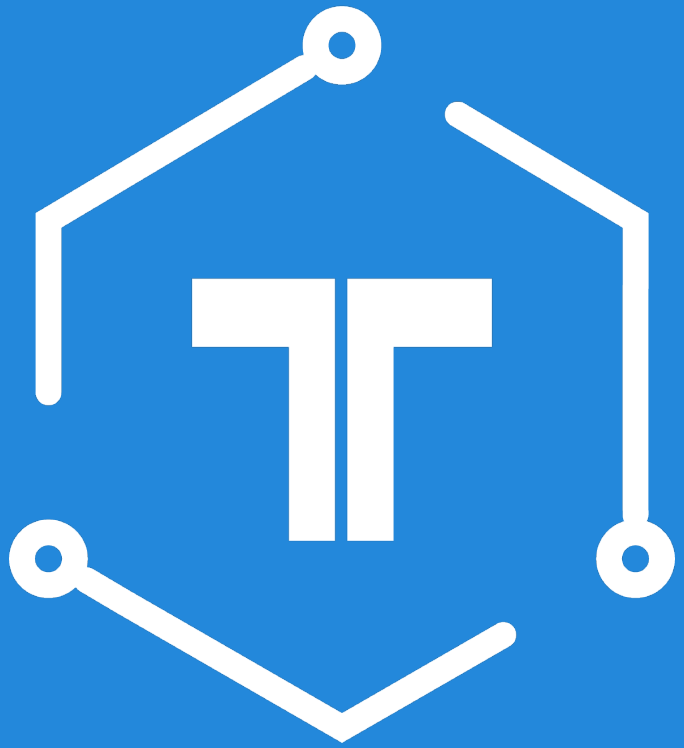
Embedded JavaScript:

The skills you have in 2014 will be all you need to make physical devices

Tim Ryan, Co-Founder / Developer

Technical Machine

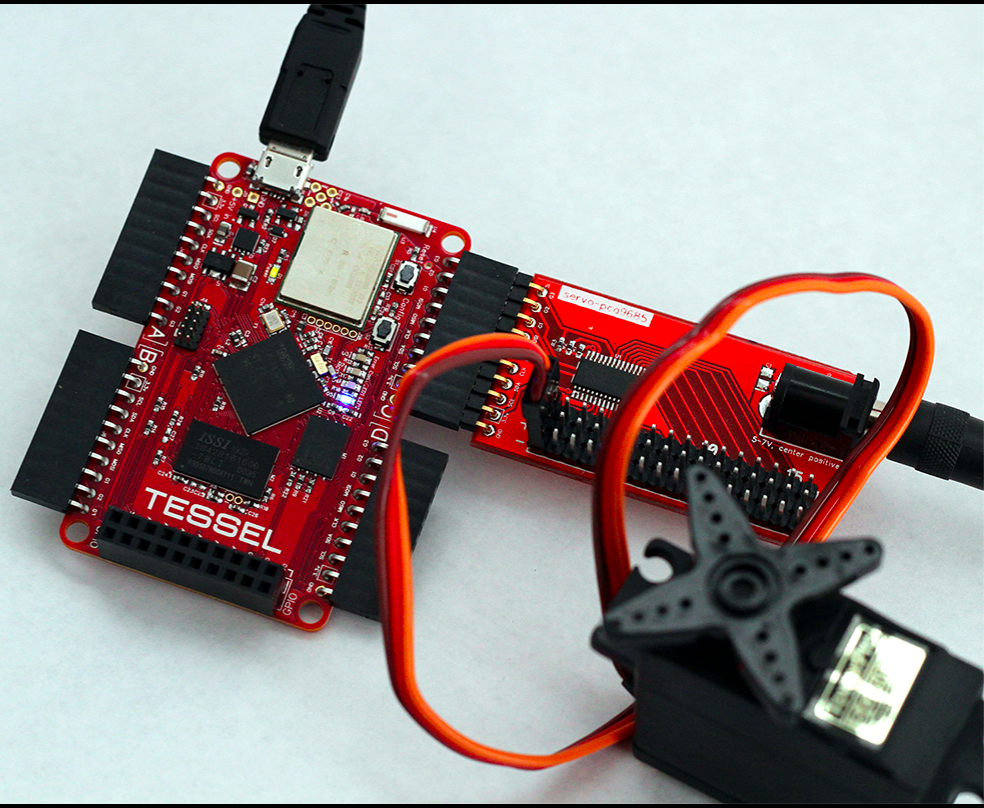


Tessel is a WiFi-enabled microcontroller that runs JavaScript.

Tim Ryan, Co-Founder / Software Developer

Technical Machine

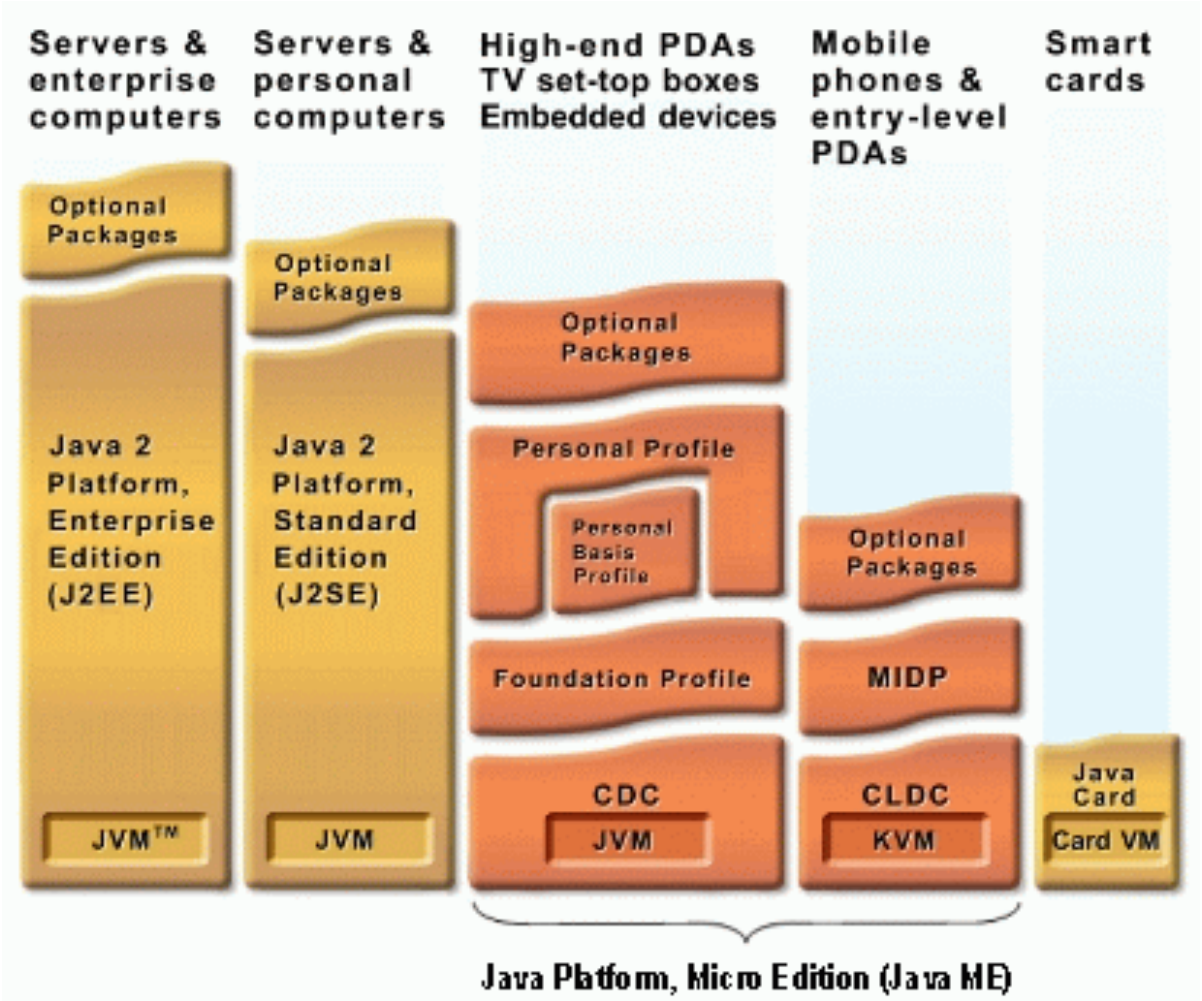
```
var tessel = require('tessel');
var servos = require('servo-pca9685')
  .connect(tessel.port('A'));

var degrees = 0;
setInterval(function () {
  servos.moveServo(1, degrees);
  degrees = degrees == 0 ? 180 : 0;
}, 500);
```



Embedded JavaScript

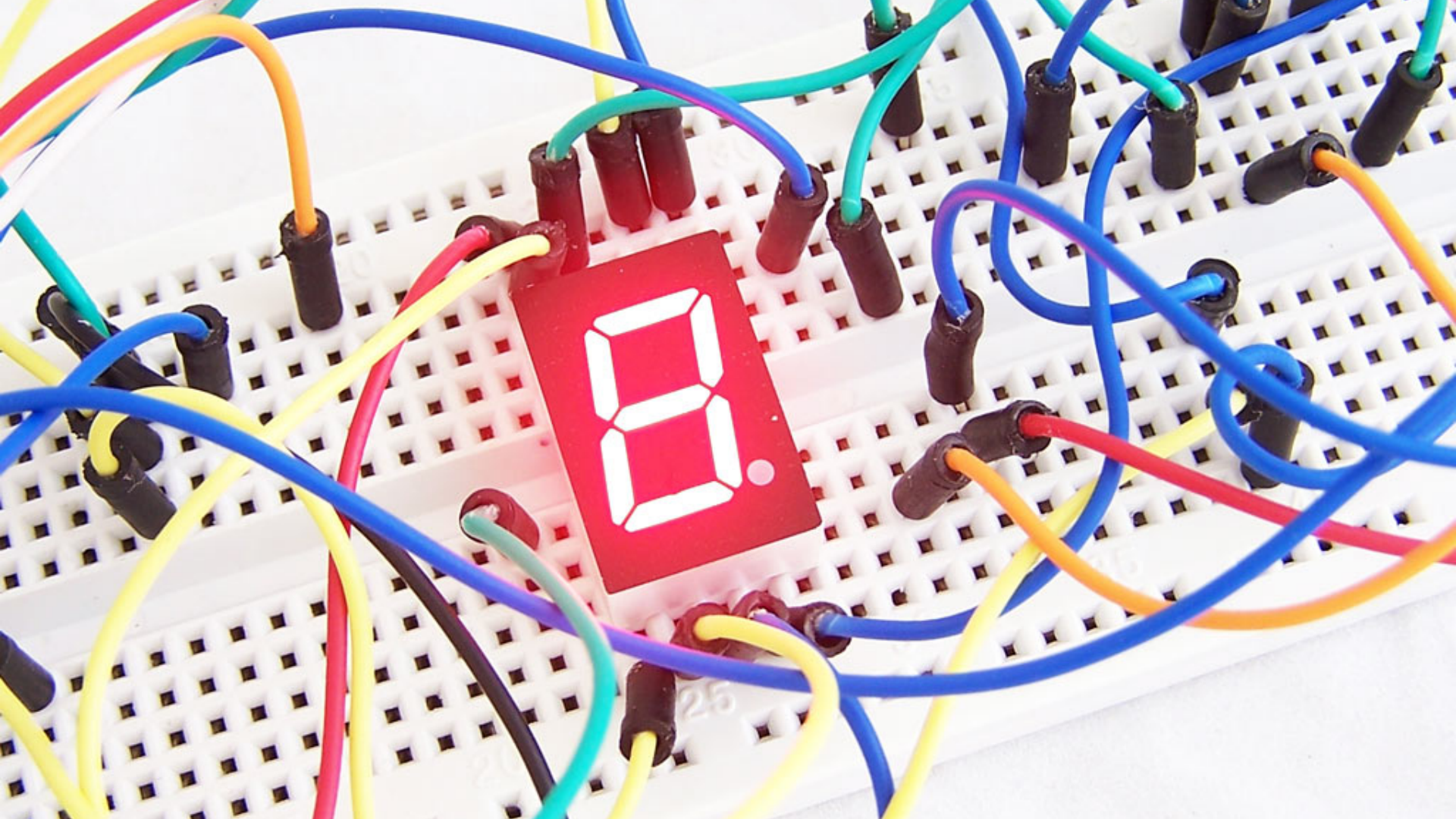

JAVA
SUN MICROSYSTEMS


Java
1 oo
2 abc
def 3
4 ghi
5 jkl
mno 6
7 pqrs
8 tuv
wxyz 9
* +
0
.


Servers & enterprise computers
Servers & personal computers
High-end PDAs TV set-top boxes Embedded devices
Mobile phones & entry-level PDAs
Smart cards
Optional Packages
Optional Packages
Optional Packages
Optional Packages
Java 2 Platform, Enterprise Edition (J2EE)
Java 2 Platform, Standard Edition (J2SE)
Personal Profile
Personal Basis Profile
Foundation Profile
MIDP
CDC
CLDC
Java Card
JVM™
JVM
JVM
KVM
Card VM
Java Platform, Micro Edition (Java ME)

High-level != Developer-friendly

Open Source Hardware



Non-Stock

All prices are in US dollars.

Digi-Key Part Number	LPC1820FET100,551-ND
Quantity Available	0 Enter Quantity Requested
Manufacturer	NXP Semiconductors
Manufacturer Part Number	LPC1820FET100,551
Description	IC MCU 32BIT 136KB FLSH 100TFBGA
Lead Free Status / RoHS Status	Lead free / RoHS Compliant

Price Break	Unit Price	Extended Price
260	5.27000	1,370.20

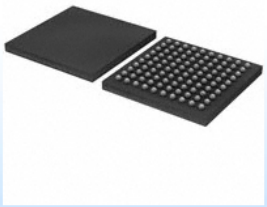
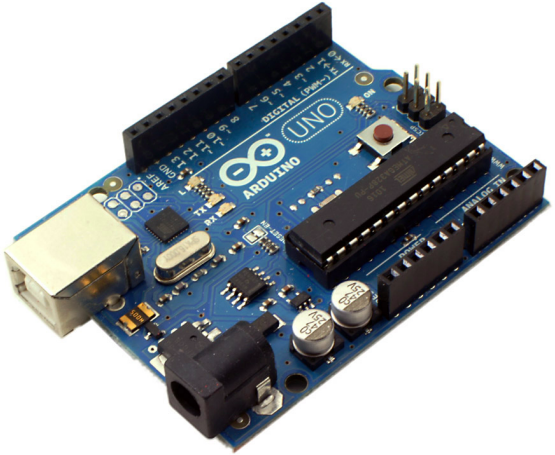


Image shown is a representation only. Exact specifications should be obtained from the product data sheet.

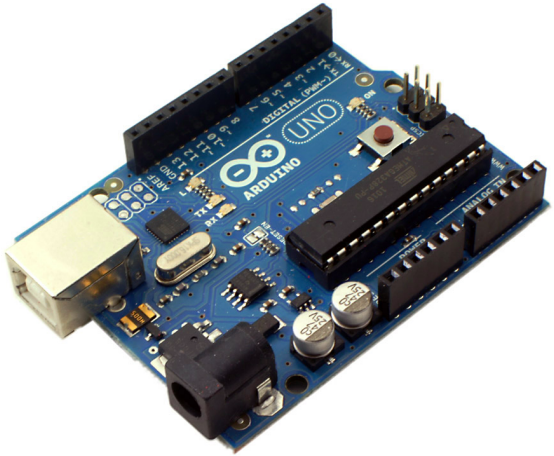
Quantity	Item Number	Customer Reference	
260	LPC1820FET100,551-ND		Add to Cart

When requested quantity exceeds displayed pricing table quantities, a lesser unit price may appear on your order. You may submit a request for quotation on quantities which are greater than those displayed in the pricing table.

Datasheets	LPC1850,30,20,10 LPC18xx User Manual
Product Photos	100-TFBGA SOT926-1
Product Training Modules	LPC1800 Series
Standard Package	260
Category	Integrated Circuits (ICs)
Family	Embedded - Microcontrollers
Series	LPC18xx
Packaging	Tray
Core Processor	ARM® Cortex™-M3
Core Size	32-Bit
Speed	150MHz
Connectivity	CAN, EBI/EMI, I²C, Microwire, SPI, SSI, SSP, UART/USART, USB OTG



ARDUINO™


UNO
ARDUINO

2005


Make:
COMPLETE PLANS »
Nuclear Fusor:
Build Your Own "Star in a Jar"
BOARD GAMES
PICK AND PLAY
A FIELD GUIDE TO MICROCONTROLLERS
+25 MAKER PROJECTS
Covert Web Router • Laser Microscope
Two-Hour Table • Desktop Foundry
Hard Cider • Easy Silk-Screen
MAKER MEDIA
makezine.com

2014

2003

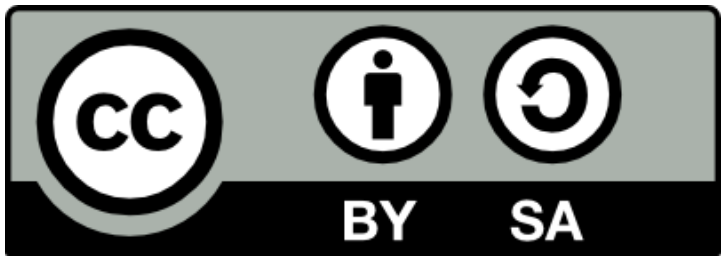
2005

2005

2003

2005

2005

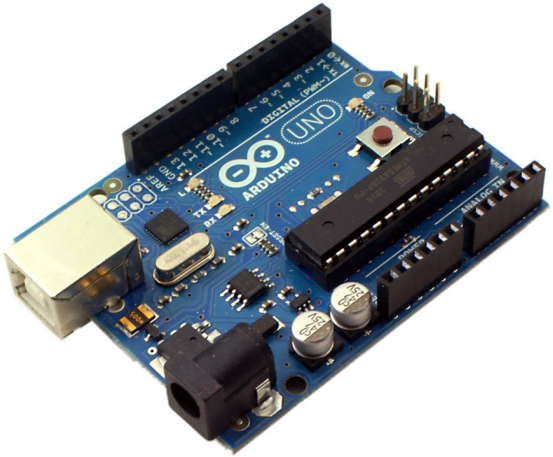


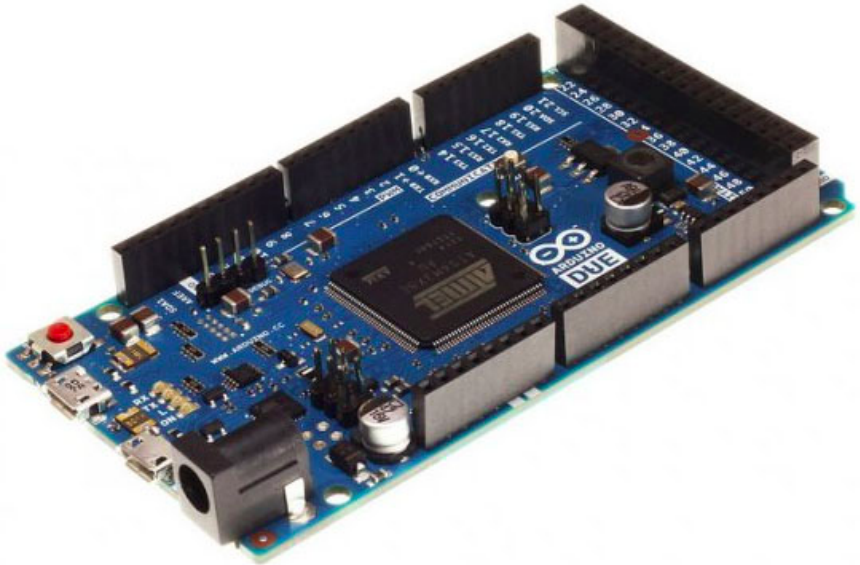
open source hardware

~2010

Microcontrollers


INVENSENSE
ITG-3200
Q6A574A
DL 1004 J
INVENSENSE
MPU-3100
Q6A465-24
DL 0943 J
INVENSENSE
IXZ-500
6111128-02
CH 0852 A





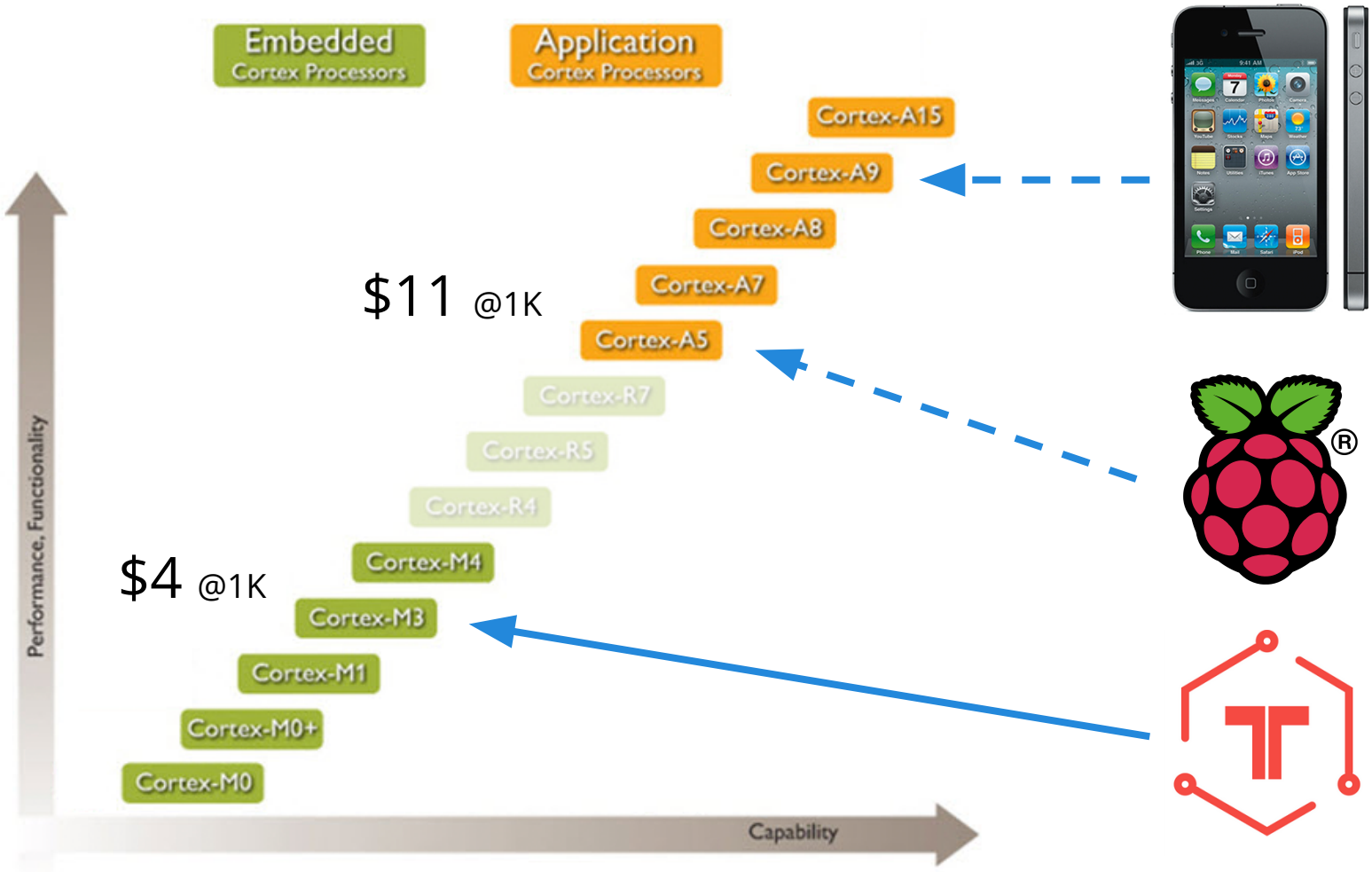
8-bit

32-bit

We can today

afford the processing power

to make developers' lives easier.


Embedded
Cortex Processors
Application
Cortex Processors
Cortex-A15
Cortex-A9
Cortex-A8
Cortex-A7
Cortex-A5
Cortex-R7
Cortex-R5
Cortex-R4
Cortex-M4
Cortex-M3
Cortex-M1
Cortex-M0+
Cortex-M0
$11 @1K
$4 @1K
Performance, Functionality
Capability


shutterstock

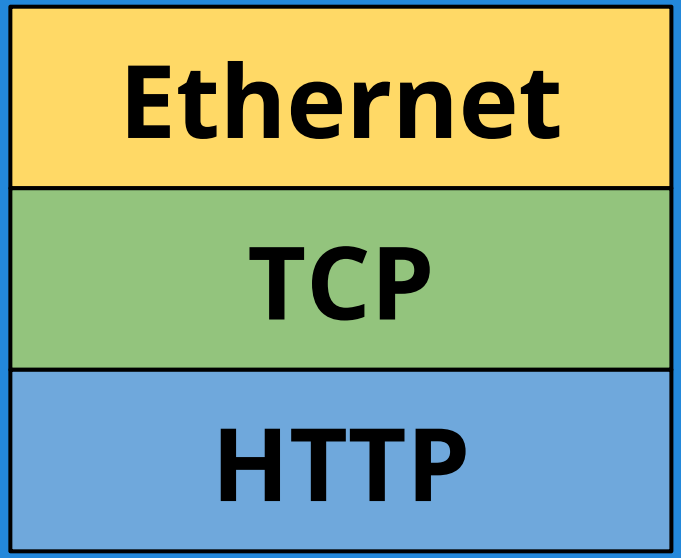

SPEED & ENDURANCE
FUJITSU
MB9AF314N
ARM
FUJITSU
MB9AF314M
ARM
FUJITSU
MB9AF314L
ARM
FM3

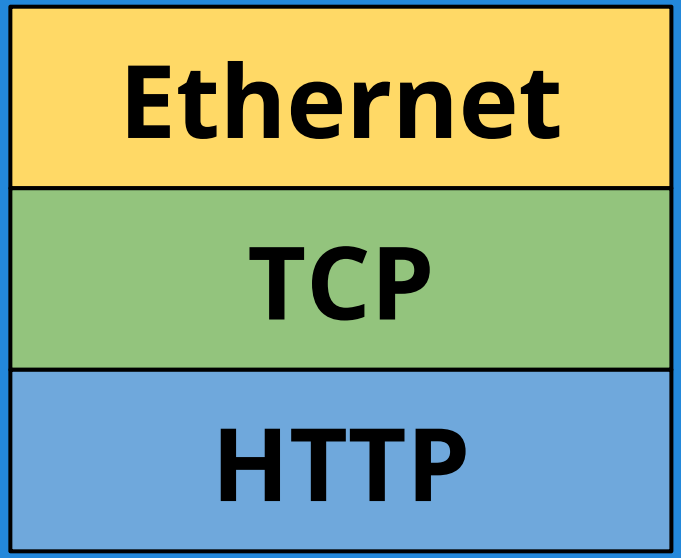
Why JavaScript?

JavaScript (and asynchronous coding) is the perfect embedded language!

JavaScript (and asynchronous coding) is the perfect embedded language!

—*Sylvia Plath*

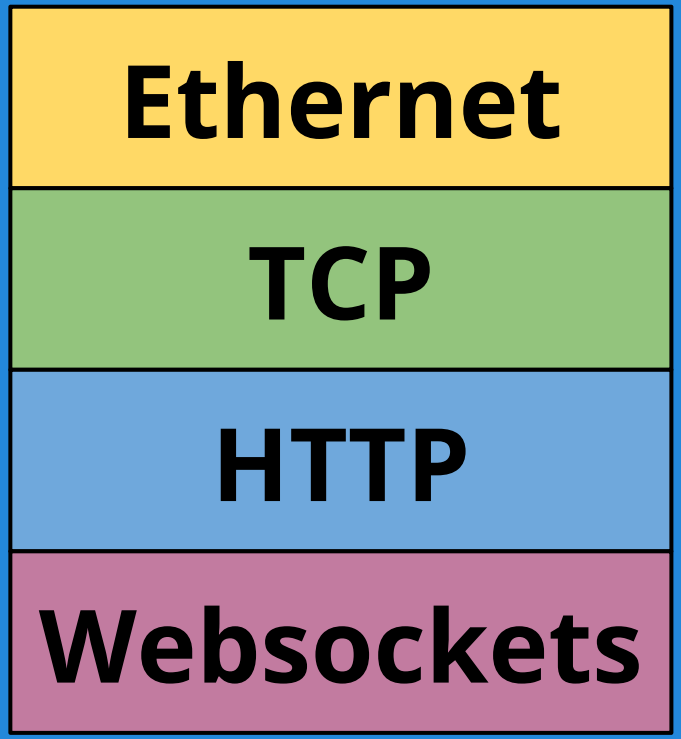

Ethernet
TCP
HTTP



(Packet-based)

(Stream-based)

(Packet-based)

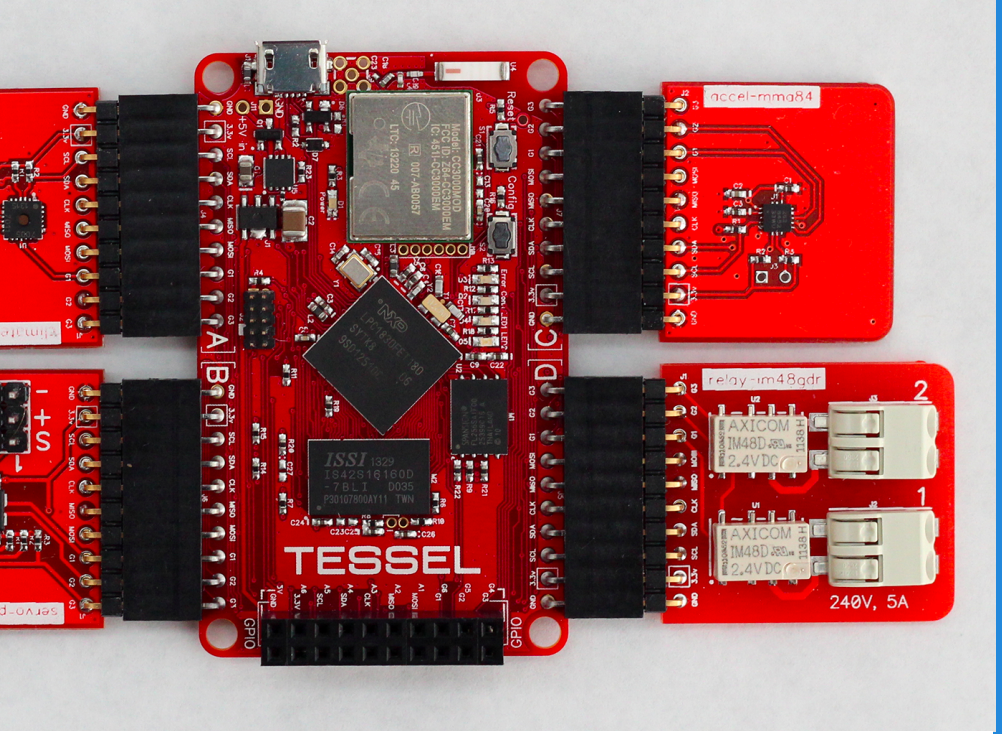


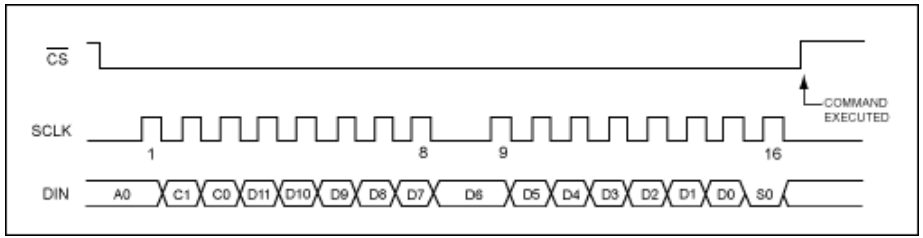
(Packet-based)

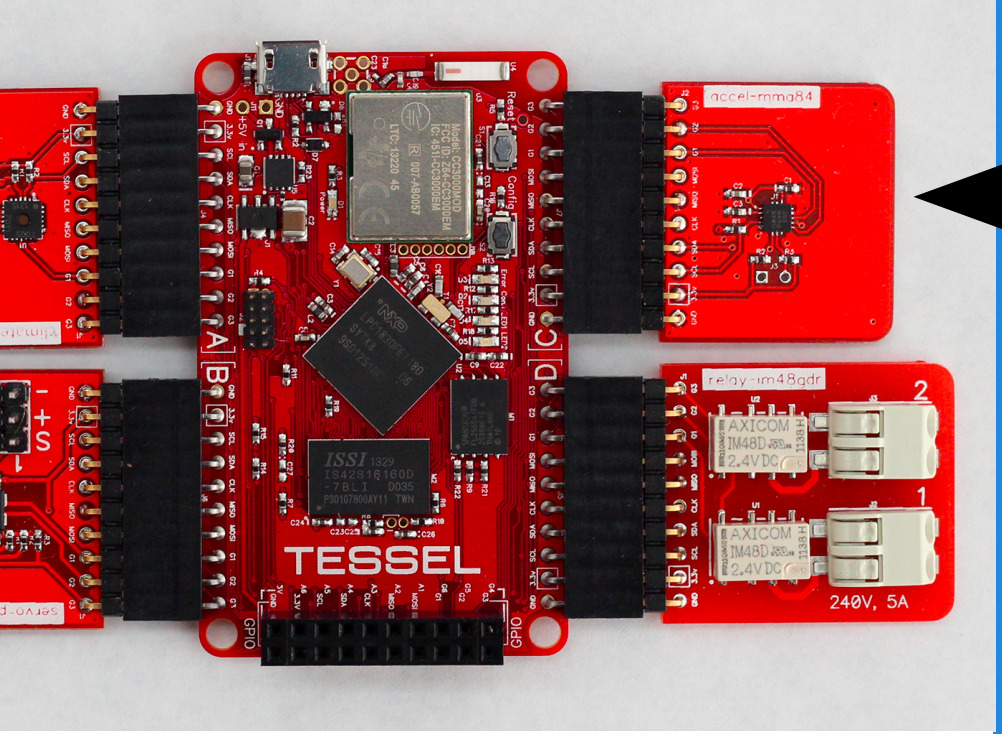
(Stream-based)

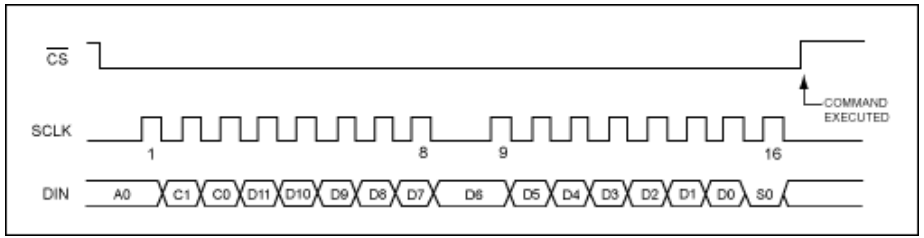
(Packet-based)

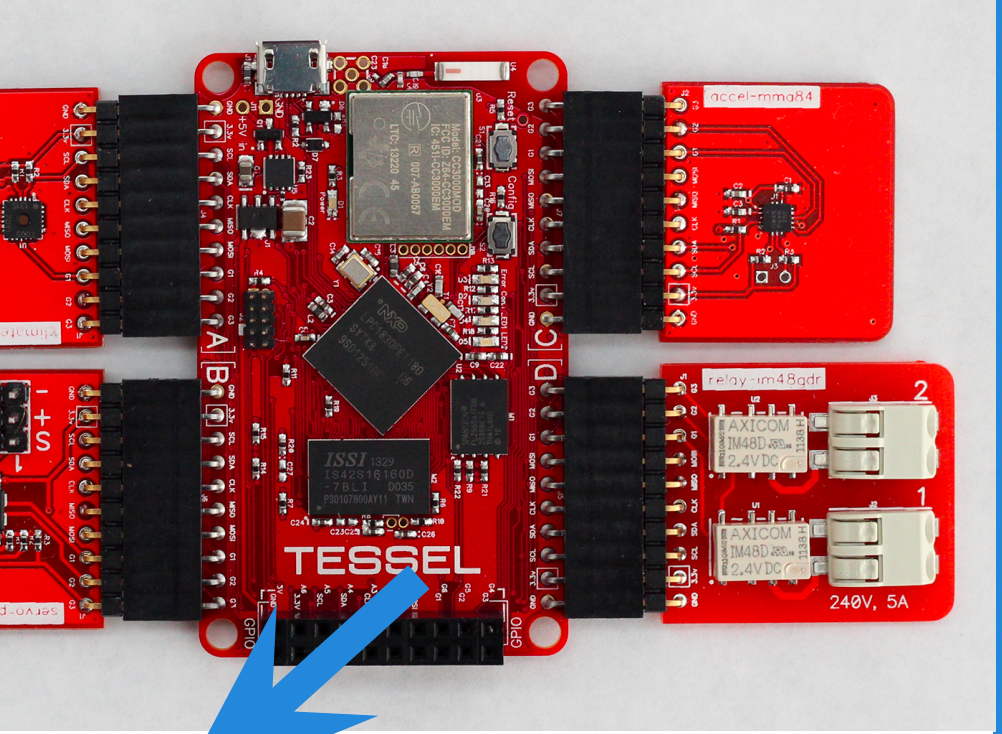
(Streaming...)

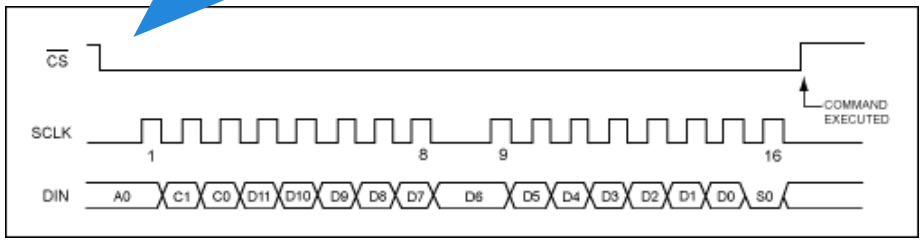

accel-mma84
relay-im48gdr
TESSEL
240V, 5A
GPIO
GPIO

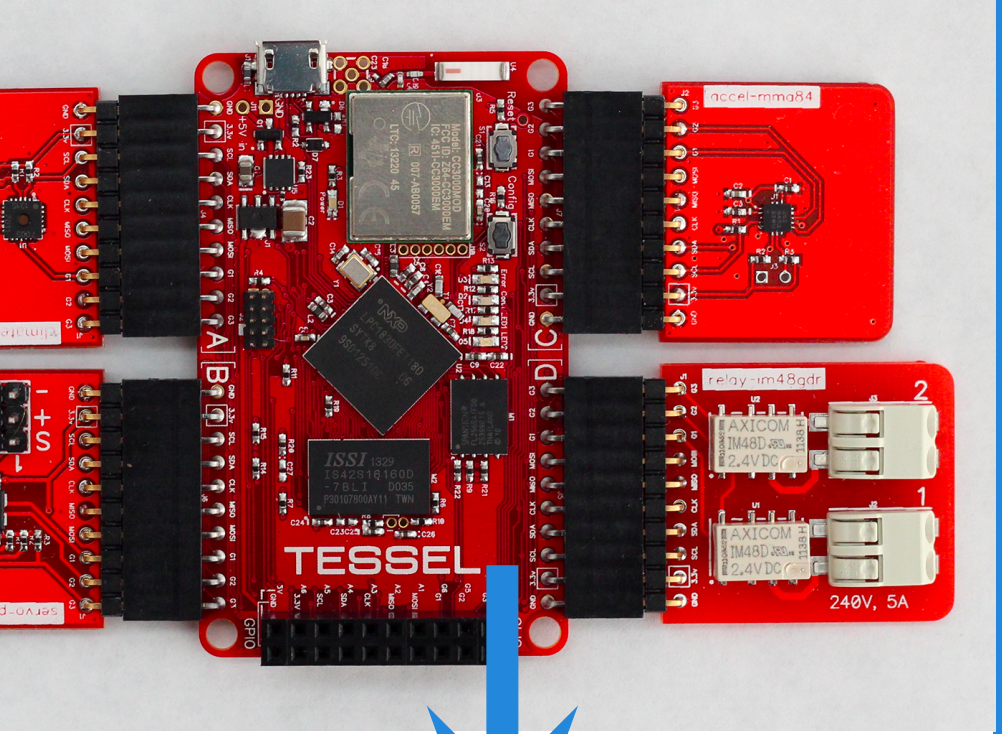

CS
COMMAND EXECUTED
SCLK
1
8
9
16
DIN
A0
C1
C0
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
S0

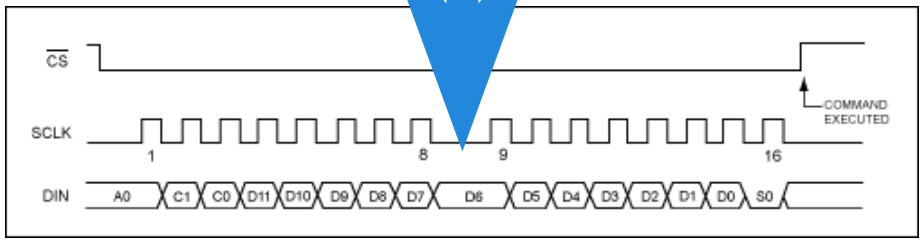

accel-mma84
relay-im48gdr
TESSEL
ISSI
AXICOM
IM48D
2.4VDC
240V, 5A
GPIO
GPIO

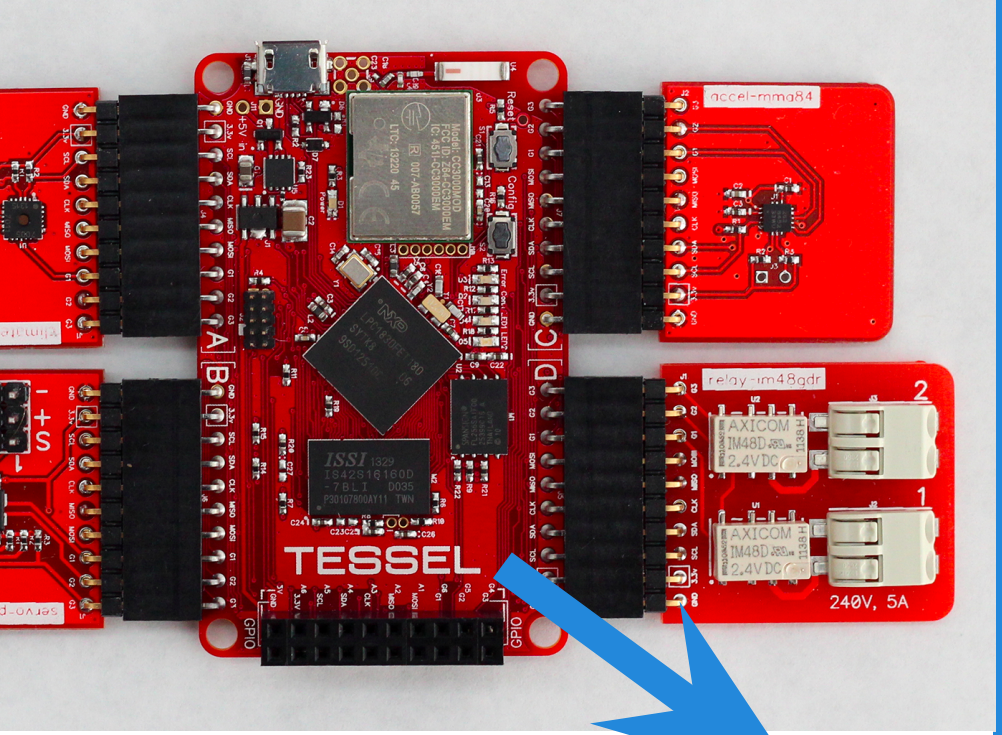

CS
SCLK
DIN
COMMAND EXECUTED
1
8
9
16
A0
C1
C0
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
S0

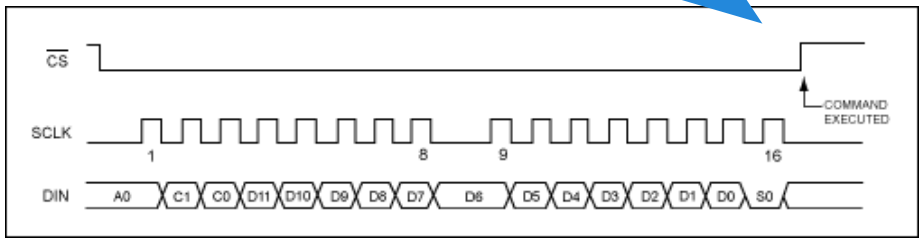

accel-mma84
relay-im48gdr
AXICOM
IM48D
2.4VDC
240V, 5A
TESSEL
ISSI
IS42S16160D
Reset
Config
GPIO

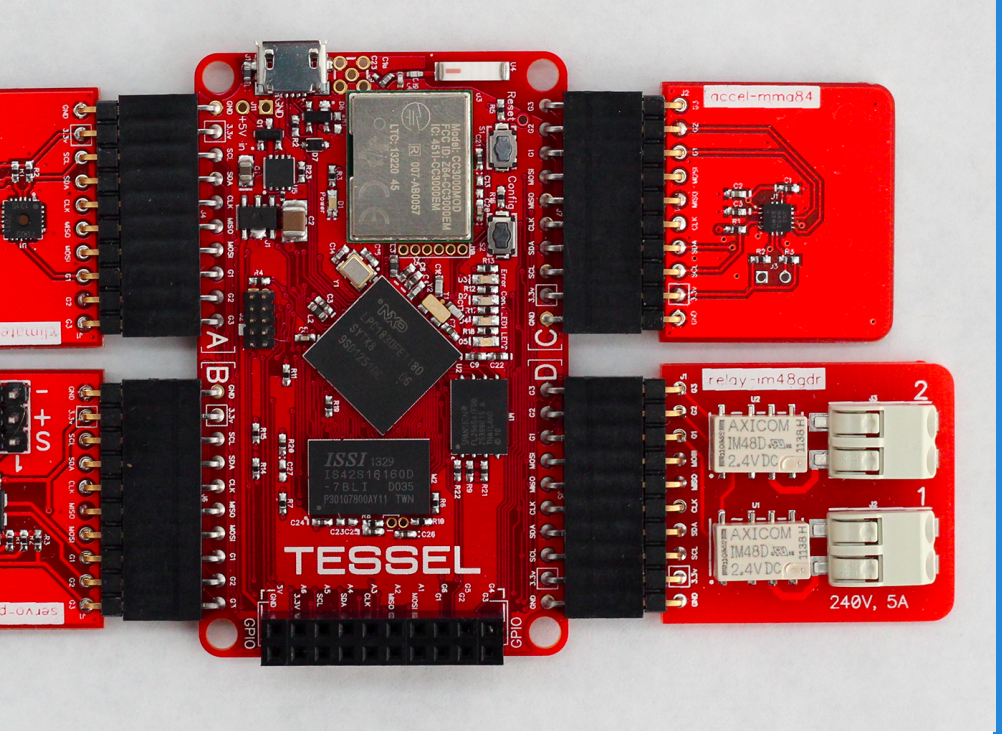

CS
COMMAND EXECUTED
SCLK
1
8
9
16
DIN
A0
C1
C0
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
S0


accel-mma84
relay-im48gdr
TESSEL
AXICOM
IM48D
2.4VDC
240V, 5A
GPIO

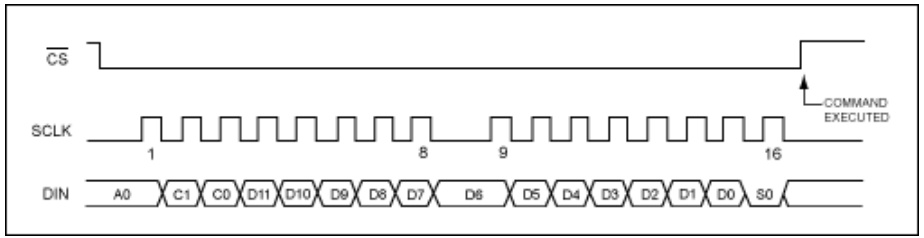

CS
SCLK
DIN
COMMAND EXECUTED
1
8
9
16
A0
C1
C0
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
S0

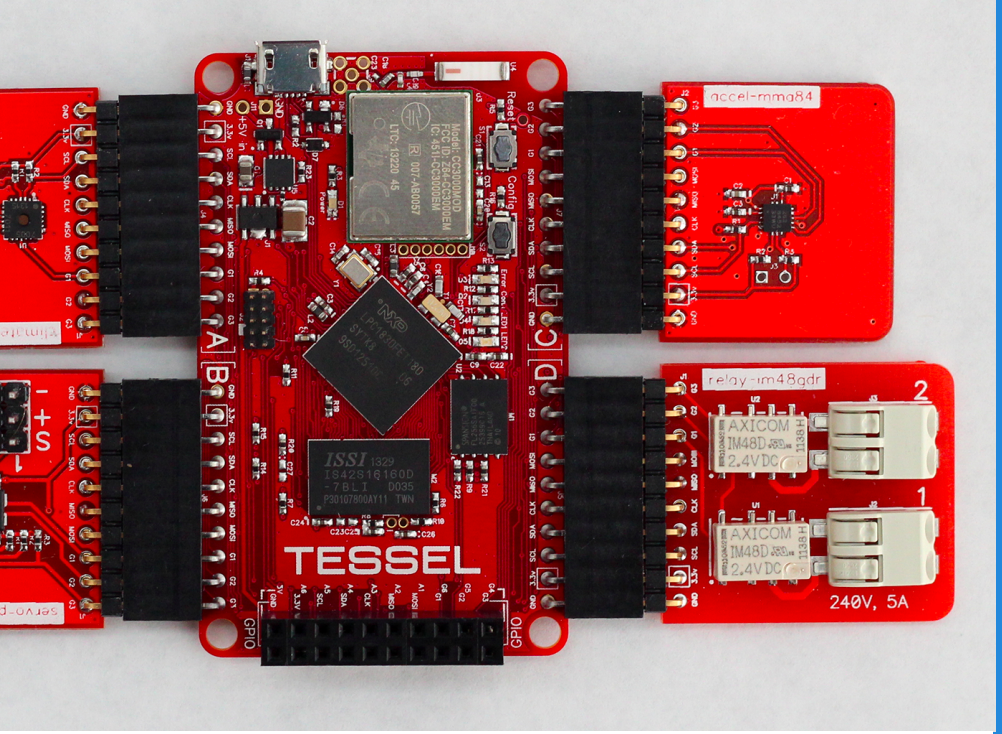

accel-mma84
relay-im48gdr
TESSEL
AXICOM
IM48D
2.4VDC
240V, 5A
GPIO
GPIO

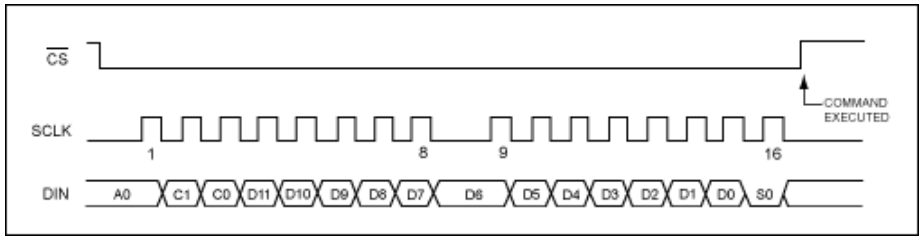

CS
COMMAND EXECUTED
SCLK
1
8
9
16
DIN
A0
C1
C0
D11
D10
D9
D8
D7
D6
D5
D4
D3
D2
D1
D0
S0



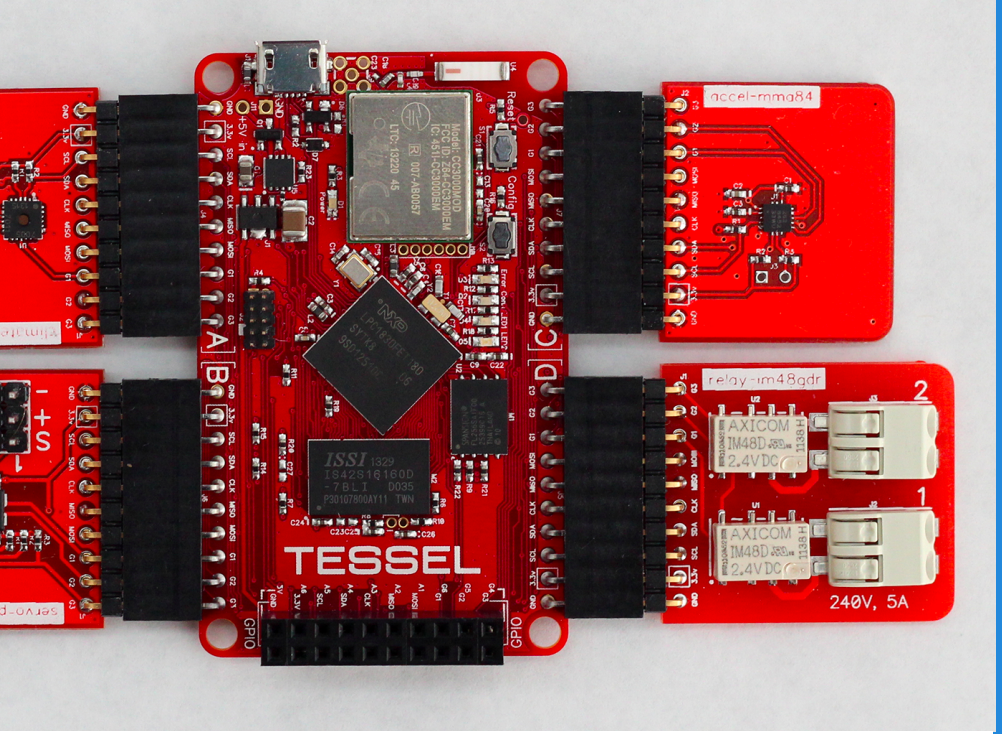

```
def receive_packet():
    buf = sock.recv()
    return buf


while True:
    print(receive_packet())
```

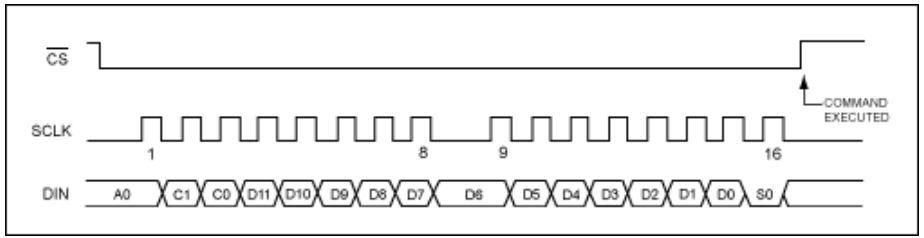






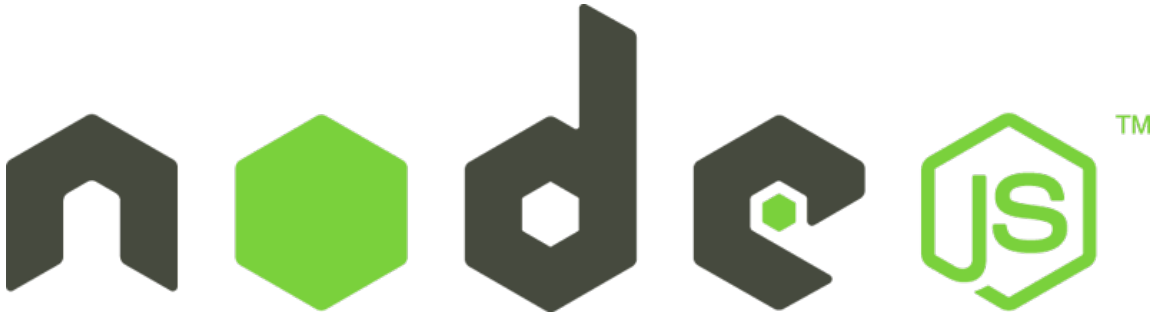

```
Future<byte[]> p =
pool.submit(new Callable<byte[]>() {
    public byte[] call() {
        return socket.recv();
    }
}

// ... while waiting ...

p.get();
```






```
socket.on('data',
function (data) {
    // handle data buffer
});

// ... other code ...
```

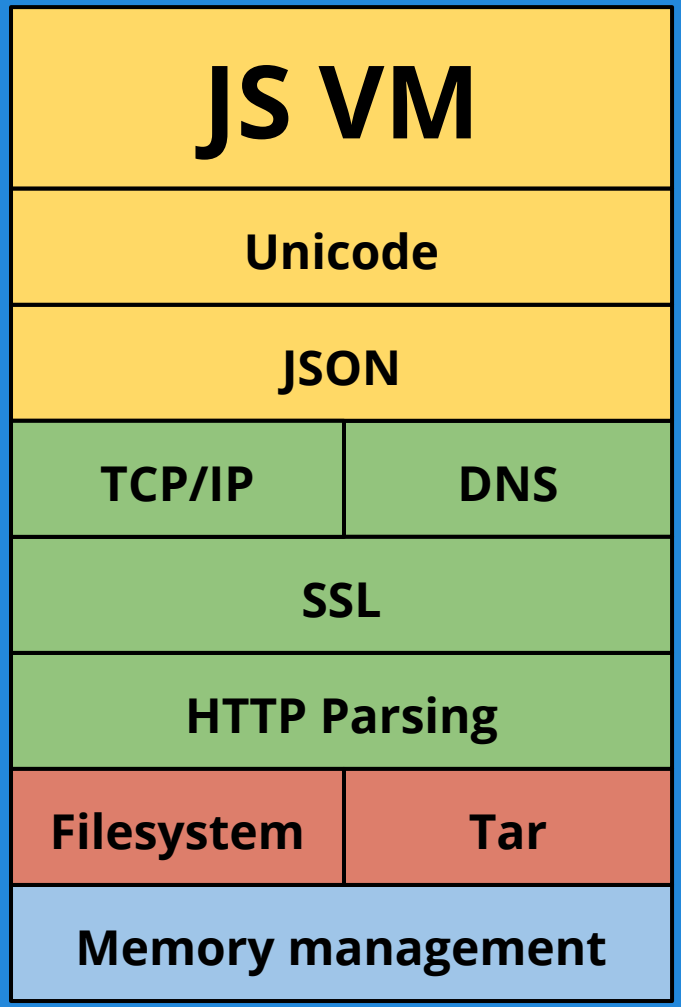
Inventing the Universe

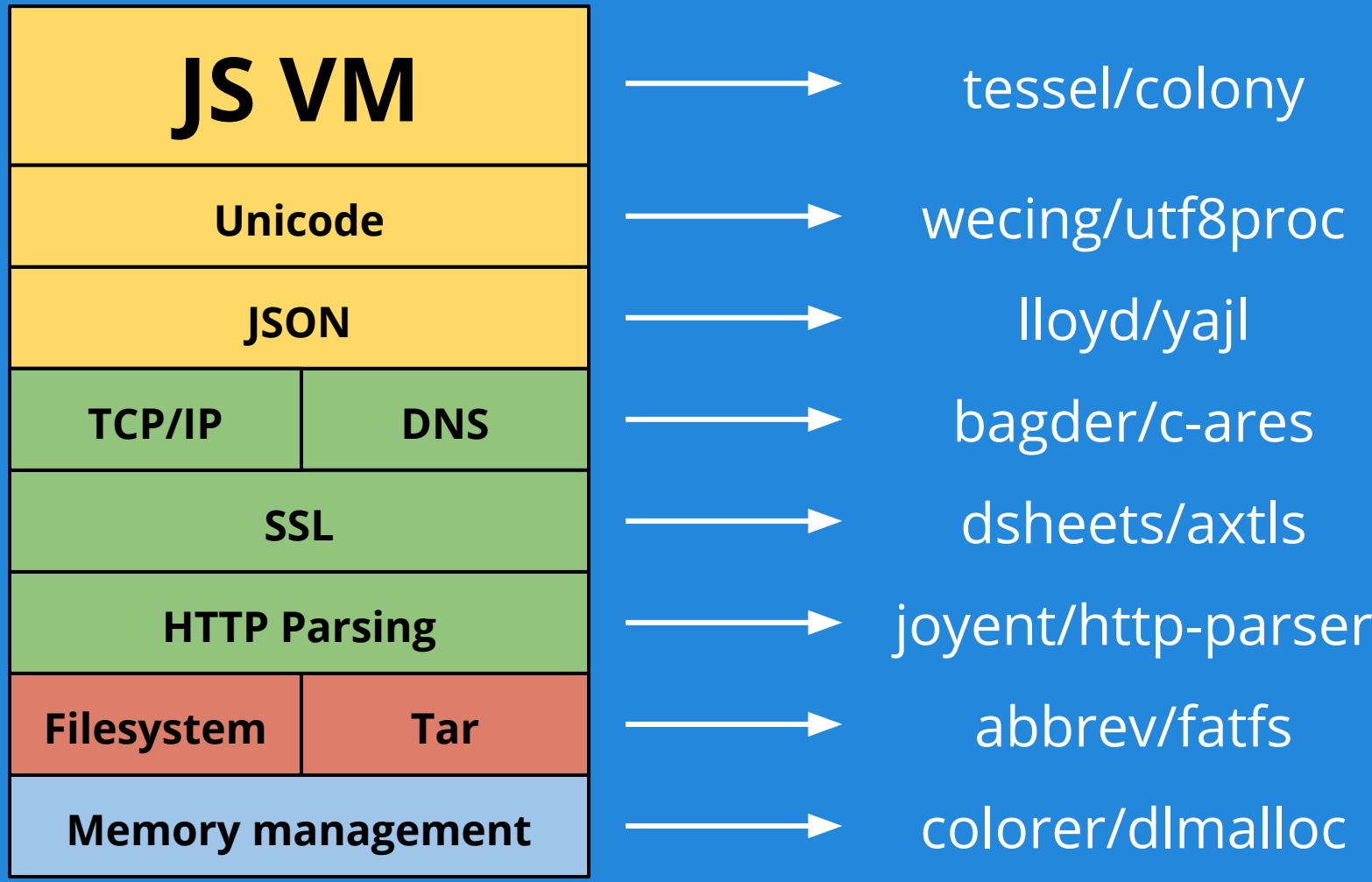

node
JS
TM

- Chrome’s JS engine
- C++
- ~10mb memory required for each instance
- POSIX/Win32 environment

- Embeddable language
- Written in C
- ~30kb memory
- Highly portable

- Embeddable language
- Written in C
- ~30kb memory
- Highly portable
- 4x-60x slower

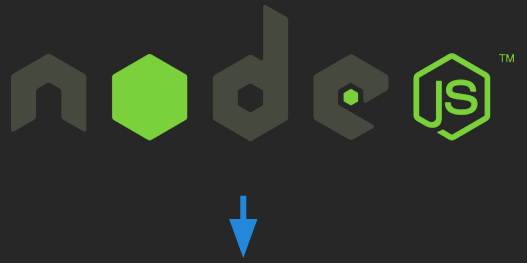

JS VM
Unicode
JSON
TCP/IP
DNS
SSL
HTTP Parsing
Filesystem
Tar
Memory management


JS VM
Unicode
JSON
TCP/IP
DNS
SSL
HTTP Parsing
Filesystem
Tar
Memory management
tessel/colony
wecing/utf8proc
lloyd/yajl
bagder/c-ares
dsheets/axtls
joyent/http-parser
abbrev/fatfs
colorer/dlmalloc

(When reinventing everything, reuse as much as possible.)

```
var tessel = require('tessel');

var led1 = tessel.led[1].writeSync('high')
var led2 = tessel.led[2].writeSync('low')

var i = 0;
setInterval(function () {
  console.log('Blinked', i++, 'times');
  led1.toggleSync();
  led2.toggleSync();
}, 100);
```

```js
var tessel = require('tessel');

var led1 = tessel.led[1].writeSync('high')
var led2 = tessel.led[2].writeSync('low')

var i = 0;
setInterval(function () {
  console.log('Blinked', i++, 'times');
  led1.toggleSync();
  led2.toggleSync();
}, 100);
```

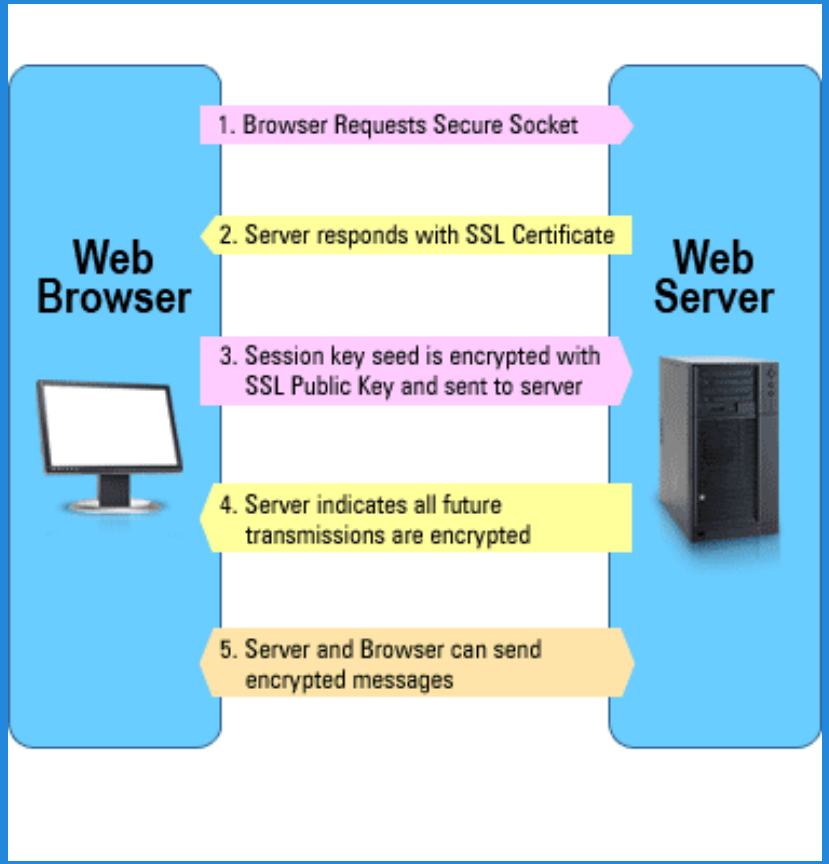



```lua
return function (_ENV, _module)
local exports, module = _module.exports, _module;

local tessel, led1, led2, i = tessel, led1, led2, i;
tessel = require(this, ("tessel"));
led1 = tessel:led((1)):output():high();
led2 = tessel:led((2)):output():low();
i = (0);
setInterval(this, (function (this)
console:log(("Blinked"), (function () local _r = i; i = _r + 1; return _r; end)(), ("times"));
led1:toggle();
led2:toggle();
end), (100));

return _module.exports;
end
```

What Comes Next

It’s becoming feasible to embed in every product a microcontroller powerful enough to run a high level language.

HTTP

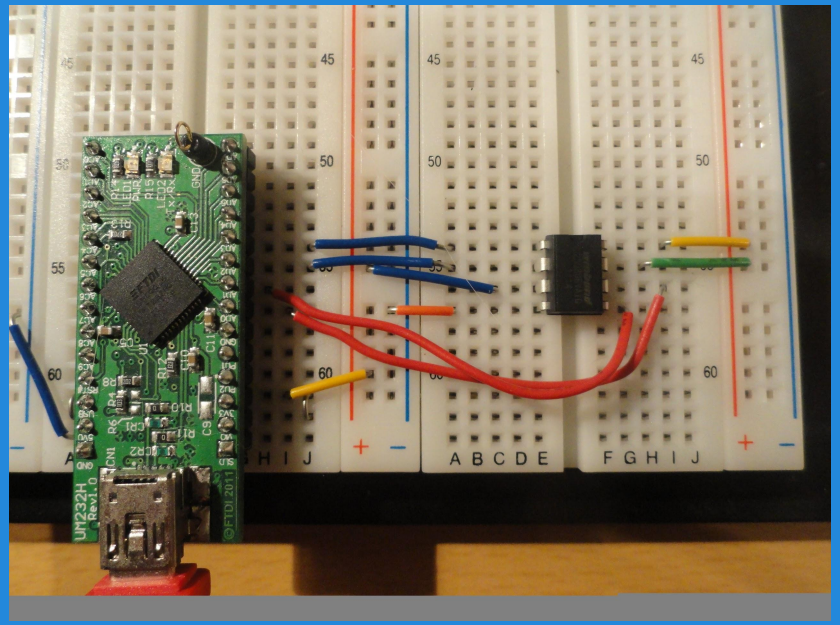

Web Browser
Web Server
1. Browser Requests Secure Socket
2. Server responds with SSL Certificate
3. Session key seed is encrypted with SSL Public Key and sent to server
4. Server indicates all future transmissions are encrypted
5. Server and Browser can send encrypted messages

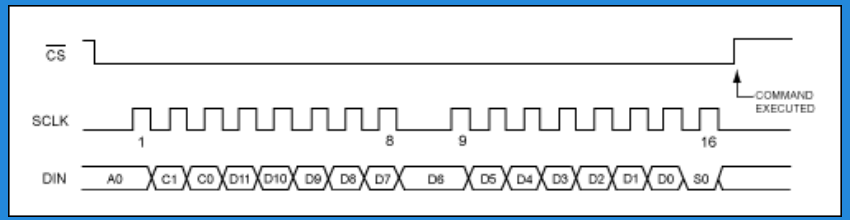
HTTP

SPI

I2C

UART





hardware

Search Results

hardware

0.0.1 by tcr

Wait for it...

hardware-resolve

0.1.1 by tcr

Resolves hardware dependencies in package.json

hardware-firmata

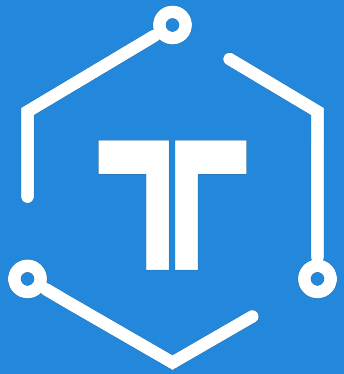
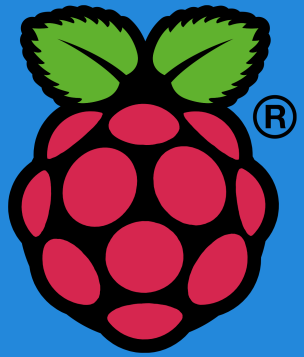
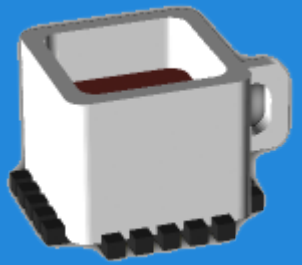
0.0.1 by tcr

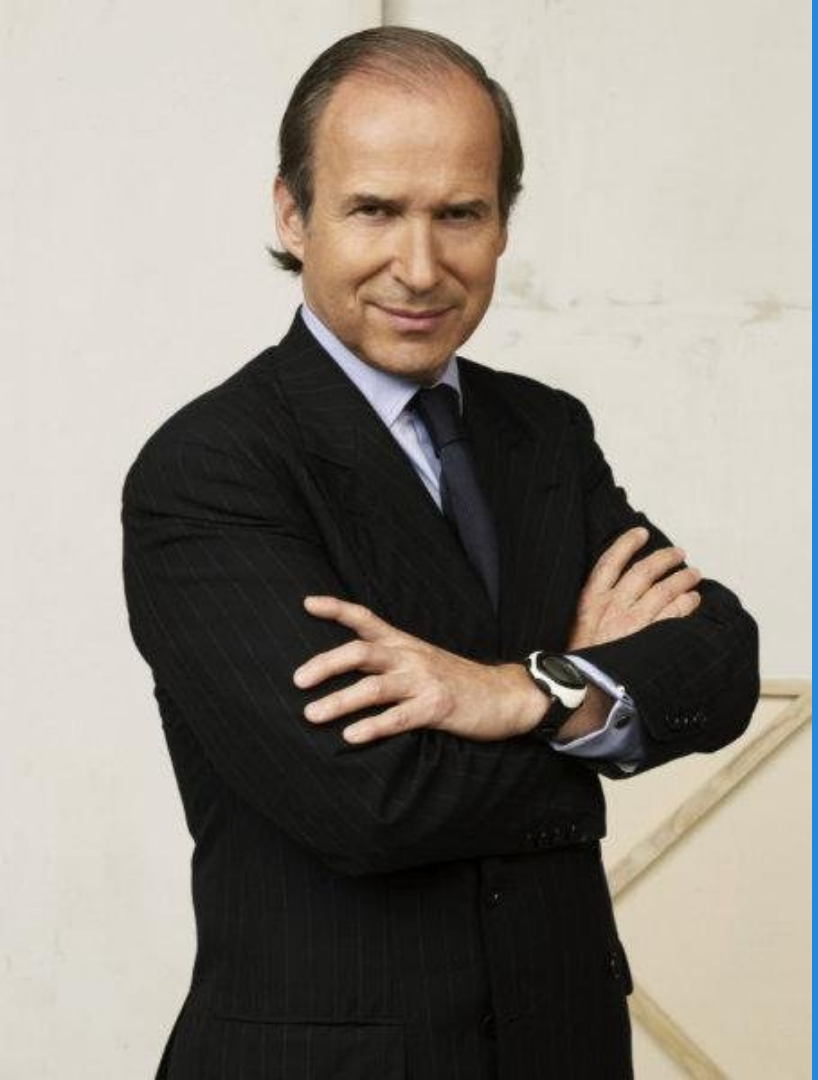
serialport-manager

0.2.3 by tmpvar

single process manager of transient serialport connections

serialport, hardware

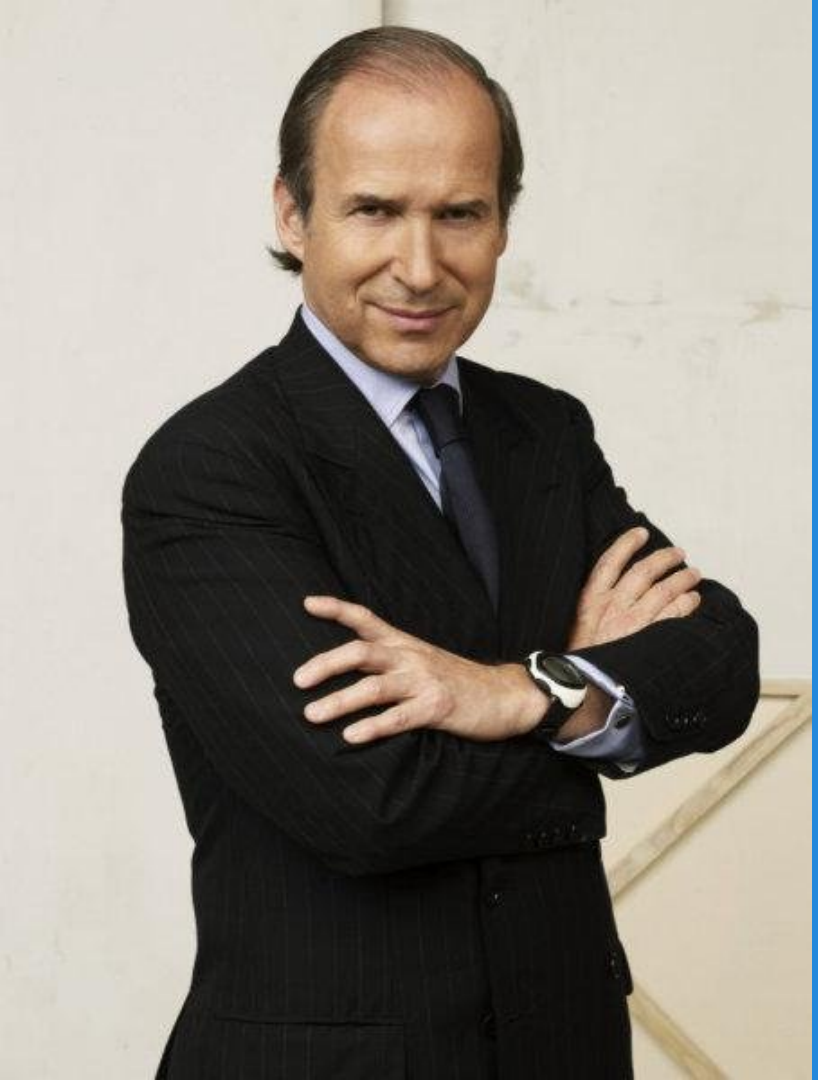







Be bold!

Be courageous!

Be amazing!

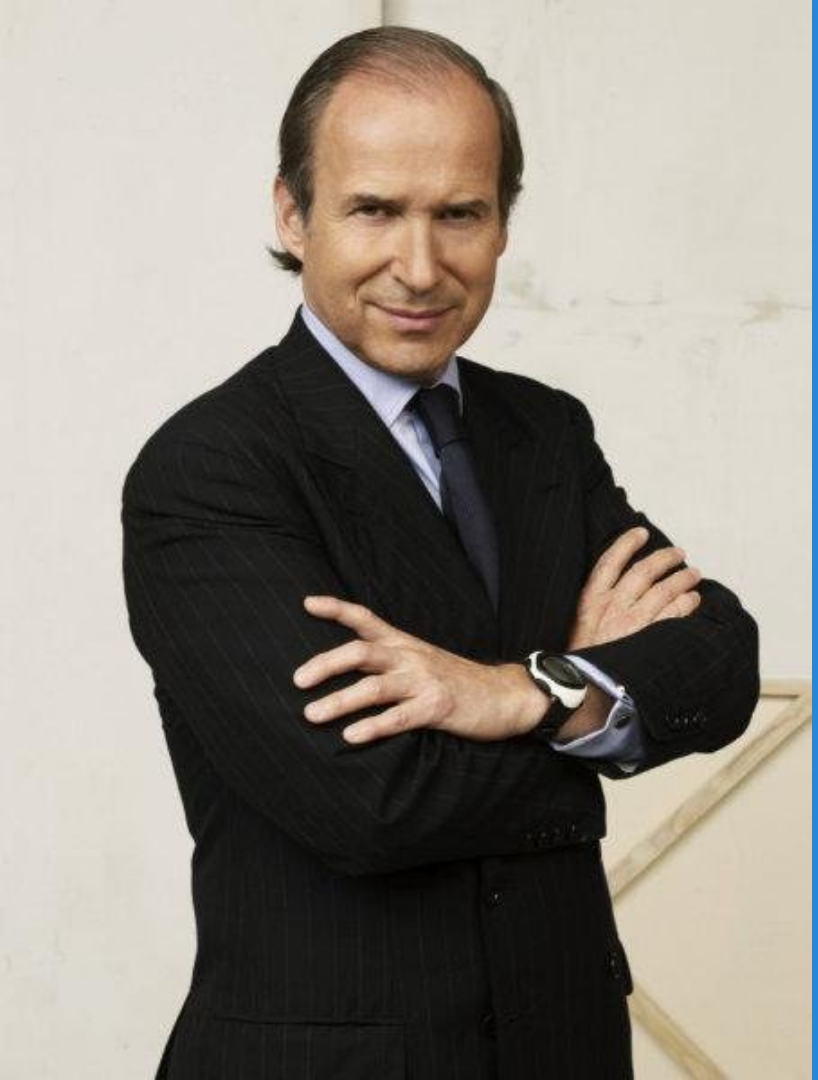


Be bold!

Be courageous!

Be amazing!

Build robots!



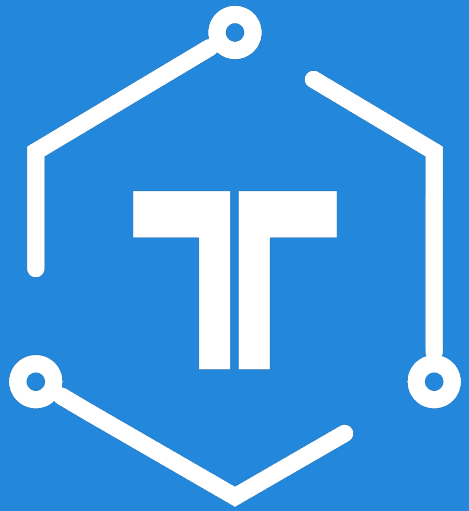
Be bold!

Be courageous!

Be amazing!

Build robots!

— Sylvia Plath



http://tessel.io

tim@technical.io

@technicalhumans