

NewTech: Getting started with Hardware

Jonathan McDowell

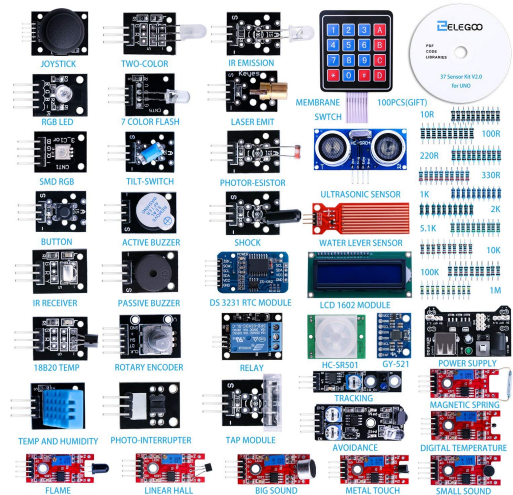
🐦 revdenoodles 🌐 noodles 🏠 www.earth.li/~noodles/ ✉ noodles@earth.li

Overview

- Hardware
 - Sensors
 - Processing
- Software
- Projects

Sensors

- Lots of cheap sensors easily available
- £27 for this lot from Amazon
- Motion sensor
- Temperature/Pressure/Humidity



Raspberry Pi

- Incredible common platform
- Full blown Linux
- Lots of software options
- Lots of IO pins



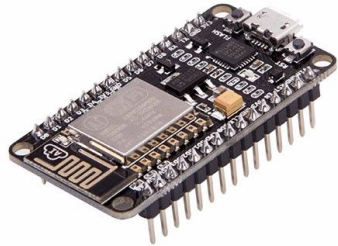
Arduino

- Kinda the original
- Atmel ATmega based (some STM32)
- Insane amounts of support
- £20
- <https://store.arduino.cc/arduino-uno-rev3>



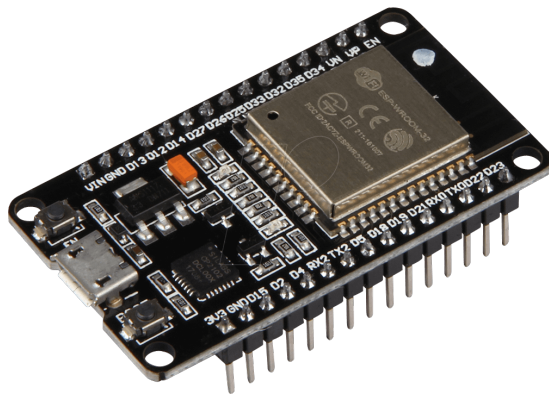
NodeMCU

- Espressif ESP8266
- Wifi!
- Lots of clones
- £5
- <https://www.nodemcu.com/>



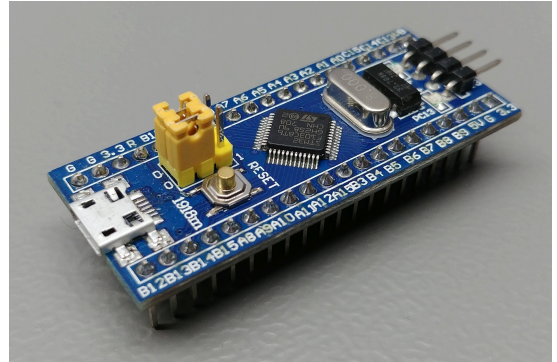
Generic ESP32

- Espressif ESP32
- Wifi AND Bluetooth (including LE)
- Not quite as mature an ecosystem as the ESP8266
- £7



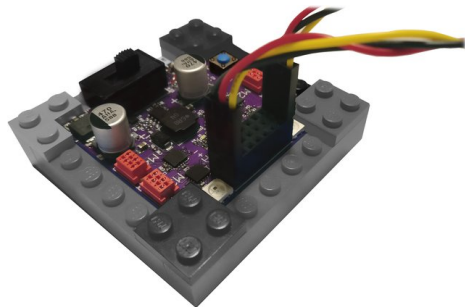
Blue Pill

- STM32F103 (ARM Cortex M3)
- Built in USB
- Bit more limited in terms of software support
- £5



Snekboard

- Microchip ATSAMD21G (ARM Cortex M0)
- Designed for Lego environments
- Motor controllers, RGB LEDs + battery support
- \$79
- <https://www.crowdsupply.com/keith-packard/snekboard>



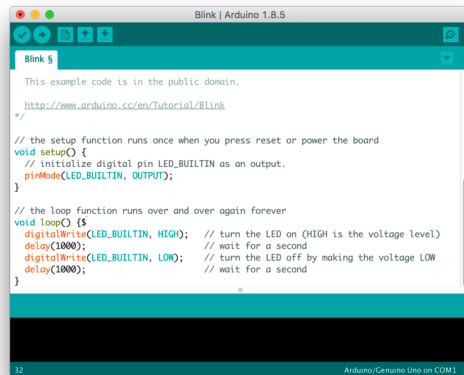
LoPy4

- ESP32 based
- LoRA/Sigfox support
- Needs an extra board to connect up USB
- £40
- <https://pycom.io/product/lopy4/>



Arduino

- C++ with lots of syntactic sugar
- Ported to lots of hardware, not just Arduino
- Impressive selection of hardware support
- <https://www.arduino.cc/>

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.8.5". The menu bar includes "File", "Edit", "Tools", and "Help". The toolbar contains icons for opening, saving, compiling, and uploading. The main text area shows the "Blink" example code. The code includes a comment about the public domain, a link to the Arduino website, and the C++ code for the Blink example. The status bar at the bottom indicates "12" and "Arduino/Genuino Uno on COM1".

```
Blink §  
  
This example code is in the public domain.  
  
http://www.arduino.cc/en/Tutorial/Blink  
*/  
  
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

Lua

- Lightweight scripting language
- Highly embeddable
- Big in the ESP8266 world
- <https://www.lua.org/>

```
led = 3 -- NodeMCU uses a different numbering scheme
tmr.alarm(1, 1000, tmr.ALARM_AUTO, function()
  gpio.write(led, gpio.LOW)
  tmr.delay(100000)
  gpio.write(led, gpio.HIGH)
end)
```

MicroPython

- Cut down Python3 variant
- Optimised for microcontrollers
- Interactive prompt
- <https://micropython.org/>

```
from machine import Pin, I2C

# creat an I2C bus
i2c = I2C(scl=Pin('X1'), sda=Pin('X2'))

# scan for list of attached devices
dev_list = i2c.scan()

# write to and read from a device
i2c.writeto(0x42, b'\x4')
data = i2c.readfrom(0x42, 4)

# memory transactions
i2c.writeto_mem(0x42, 0x12, b'\x')
data = i2c.readfrom_mem(0x42, 0x12, 2)
```

JavaScript

- JavaScript for IoT
- Embedded JS engine (XS)
- ESP8266 / ESP32 focussed at present
- Trying to produce a platform for IoT
- <https://github.com/Moddable-OpenSource/moddable>



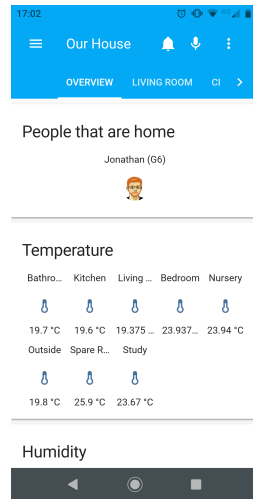
Clock

- ESP8266 module + MAX7219 dot matrix LED
- NTP over WiFi
- Prototyped in MicroPython



Home Automation

- Home Automation is my excuse to build things
- <https://www.home-assistant.io/>
- Ties all my sensors together
- Allows me to control things too



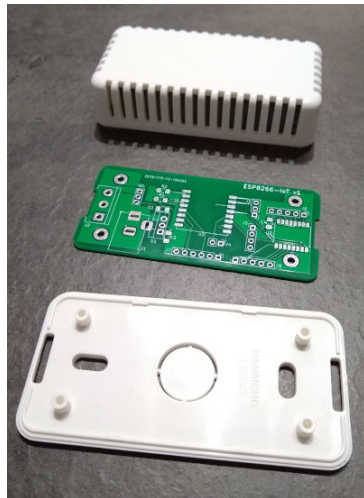
Lights

- Dimmable WS2812 LED strips, ESP8266 controller
- Code easily available for WS2812 for lots of platforms
- Some soldering required here (but possibly not)



IoT PCB

- A good example of how things can get silly
- Needed a better way to case my projects
- Kept seeing \$2 PCB offers from JLCPCB
- Designed an ESP8266 PCB using KiCad to fit a suitable case



Thank you

