

Microcontrollers and The Internet Of Things (IoT)

Khalid Baheyeldin
October 2018





Agenda



- The Internet of Things (IoT)
- Microcontrollers vs. ‘computers’
- Arduino and Arduino IDE
- ESP8266 and ESP32
- MicroPython
- Home Assistant, a brief overview
- STM32 ARM Cortex
- Projects





Internet of Things



- Means different things for different people
- Scary:
 - “Fridge will order eggs automatically”
 - Consumption habits siphoned off and recorded
 - “Hacked security cameras viewable to anyone”





Internet of Things



- Useful:
 - “Crop monitoring: soil humidity, temperature, ...”
 - Locally monitored alarm system
- Not TCP/IP, nor internet connected
 - e.g. LoRa and LoRaWAN





Microcontrollers?



- For someone who programmed computers for decades:
 - No peripherals (display, disk, network)
 - No Operating System
 - No multitasking (sort of ...)
 - No scheduling (Cron)
 - No command line (Shell), nor User Interface
 - No File System (directories, files)





What then ...



- Program stored in EEPROM
- Loop forever executing program (simple or not)
- GPIO pins for input/output
- Hardware probes for programming
 - JTAG
- Hard to debug (no user interface)
- Ubiquitous embedded systems
 - Thermostat, Oven, Water softener, ...

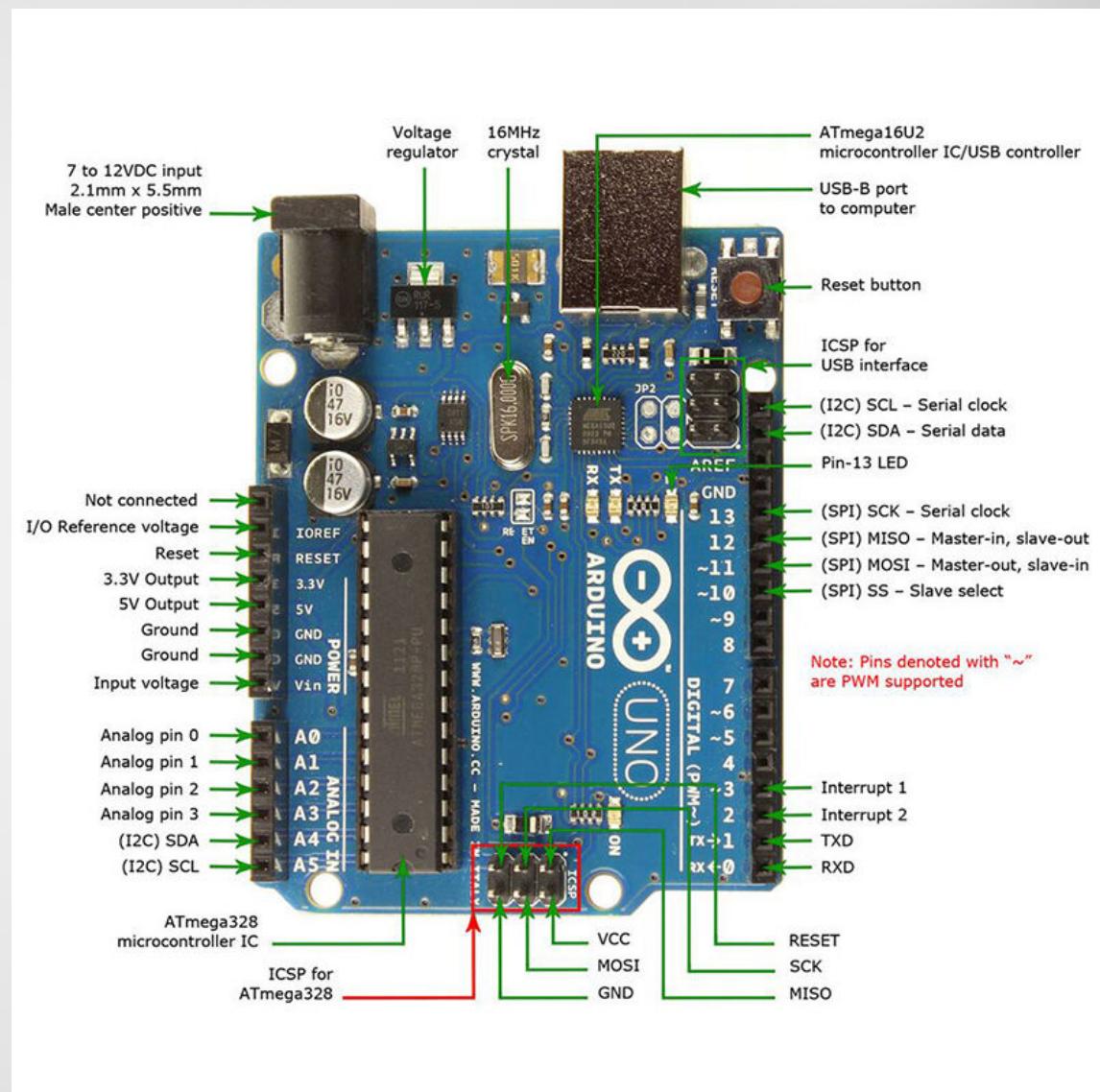




Arduino



- Microcontroller + some stuff
- GPIO Pins
- Open Source Hardware
- Ecosystem ('shields', Ethernet, motors, ...)
- Clones from China ~ \$9





Arduino Uno Cons



- Relatively slow (16MHz)
- Limited RAM (2K)
- Limited flash storage (32K)
- Large-ish size (6.8 x 5.3 cm)
 - Smaller variants available





Arduino IDE



- Open Source software
- Java application, front end to compiler, linker
- Basically C/C++, with Arduino specific libraries
- Serial Monitor, to display stuff to from your 'sketches'
- Support added for a wide variety of boards
- Libraries galore ...
- Amateur microcontroller revolution





Alternatives



There are many ...

Will focus on a few ...

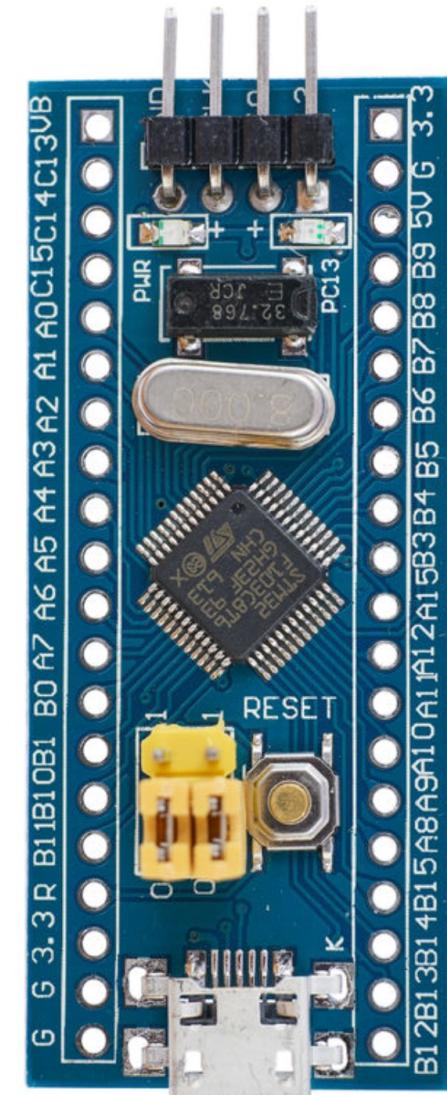




STM32 Blue Pill



- Blue Pill variant
- ARM Cortex M3 MCU
- Fast: 72MHz (4.5X Arduino Uno)
- 20 K RAM, 128 K Flash
- Small footprint (5.3 x 2.25 cm)
- Half pins 5V tolerant
- 5V power pin
- Flimsy microUSB
- Cheap: ~ \$3

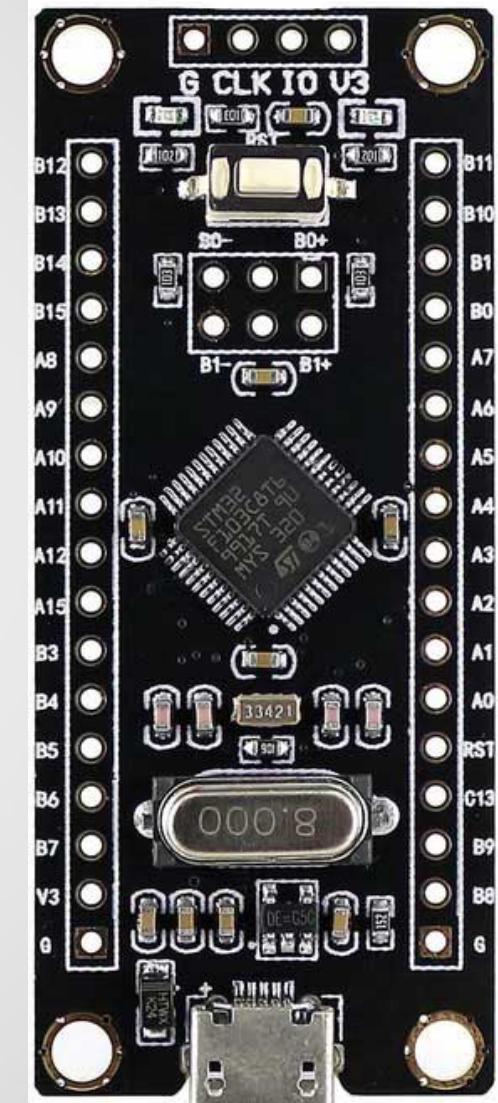




STM32 Black Pill



- Black Pill variant
- ARM Cortex M3 MCU
- Fast: 72MHz (4.5X Arduino Uno)
- 20 K RAM, 128 K Flash
- Similar foot print + mounting holes
- All pins 3.3V
- 3.3V power only (except USB)
- Sturdy microUSB
- Cheap: ~ \$3





ESP8266



- WiFi addon board for other MCUs
- Full TCP/IP stack
- RISC CPU @ 80 MHz
- 80 KB RAM, 512 KB to 4MB Flash
- GPIO pins (2 for ESP01, more for others)
- Modules: ESP01, NodeMCU, Lolin, ...
- Lua firmware, Arduino IDE, MicroPython

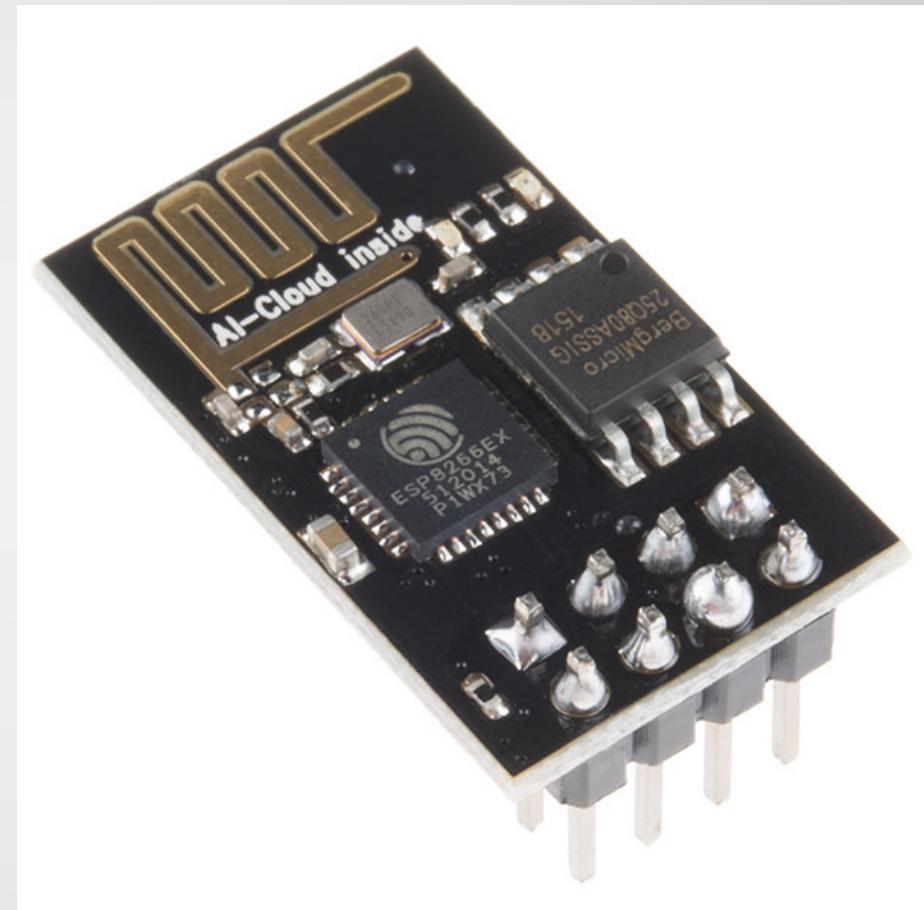




ESP01



- Smallest ESP8826
- Minimal WiFi addon module
- 2 GPIO pins
- Not breadboardable (requires adapter)
- ~ \$3

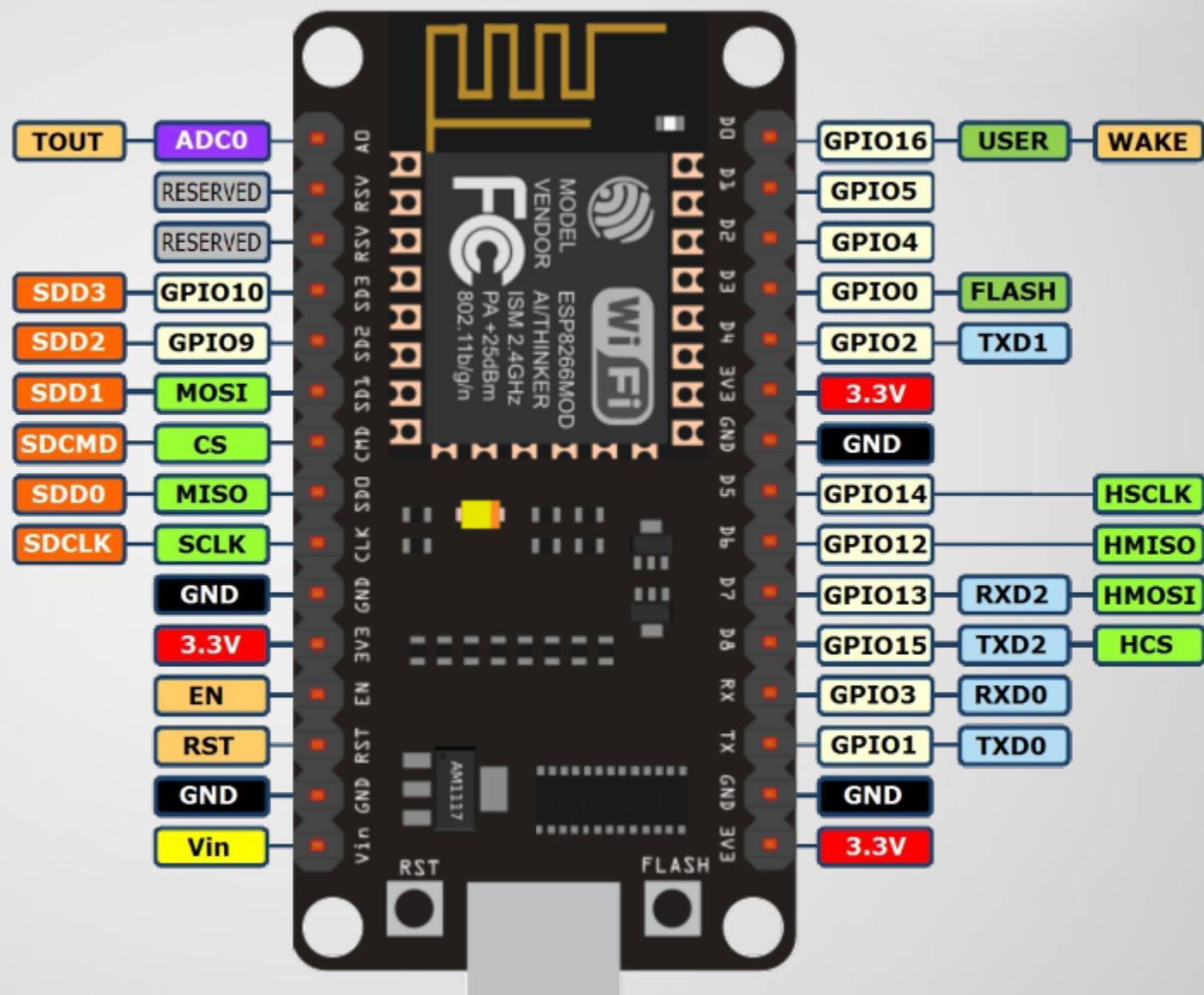




NodeMCU

2bits

- ESP8266
- More GPIOs
- Larger footprint
- MicroUSB
- ~ \$5





ESP32



- WiFi and Bluetooth addon module
- Full TCP/IP stack
- RISC CPU Dual/Single Core @ 160/240 MHz
- 520 KB RAM, 4 to 16MB Flash
- GPIO pins
- Modules: NodeMCU32S, Lolin32, ...etc.
- Lua firmware, Arduino IDE, MicroPython





MicroPython



- Python for microcontrollers
- REPL (command prompt: Read Evaluate Print Loop)
- No compiler needed!
- Originally for PyBoard, WiPy
 - Expensive
- Ported to ESP8826 and ESP32
 - Low cost + WiFi





MicroPython



- Python for microcontrollers
- REPL (command prompt: Read Evaluate Print Loop)
- No compiler needed!
- Originally for PyBoard, WiPy
 - Expensive
- Ported to ESP8826 and ESP32
 - Low cost + WiFi





Applications



- Solar powered Weather Station
 - Parts ordered, code written, with MCU sleep
 - Idea scrapped, due to weather proofing
- Indoor Temperature and Humidity Sensor
 - ESP8266 + DHT22
- Garage Door Sensor
 - ESP32
 - DHT22
 - 2 X HC-SR05 Ultrasonic distance sensor
 - Light sensor





Applications



- MQTT
 - Message Broker, used for IoT
- Home Assistant
 - Python Home Automation package
 - Lots of supported devices





MicroPython



Demo





Break

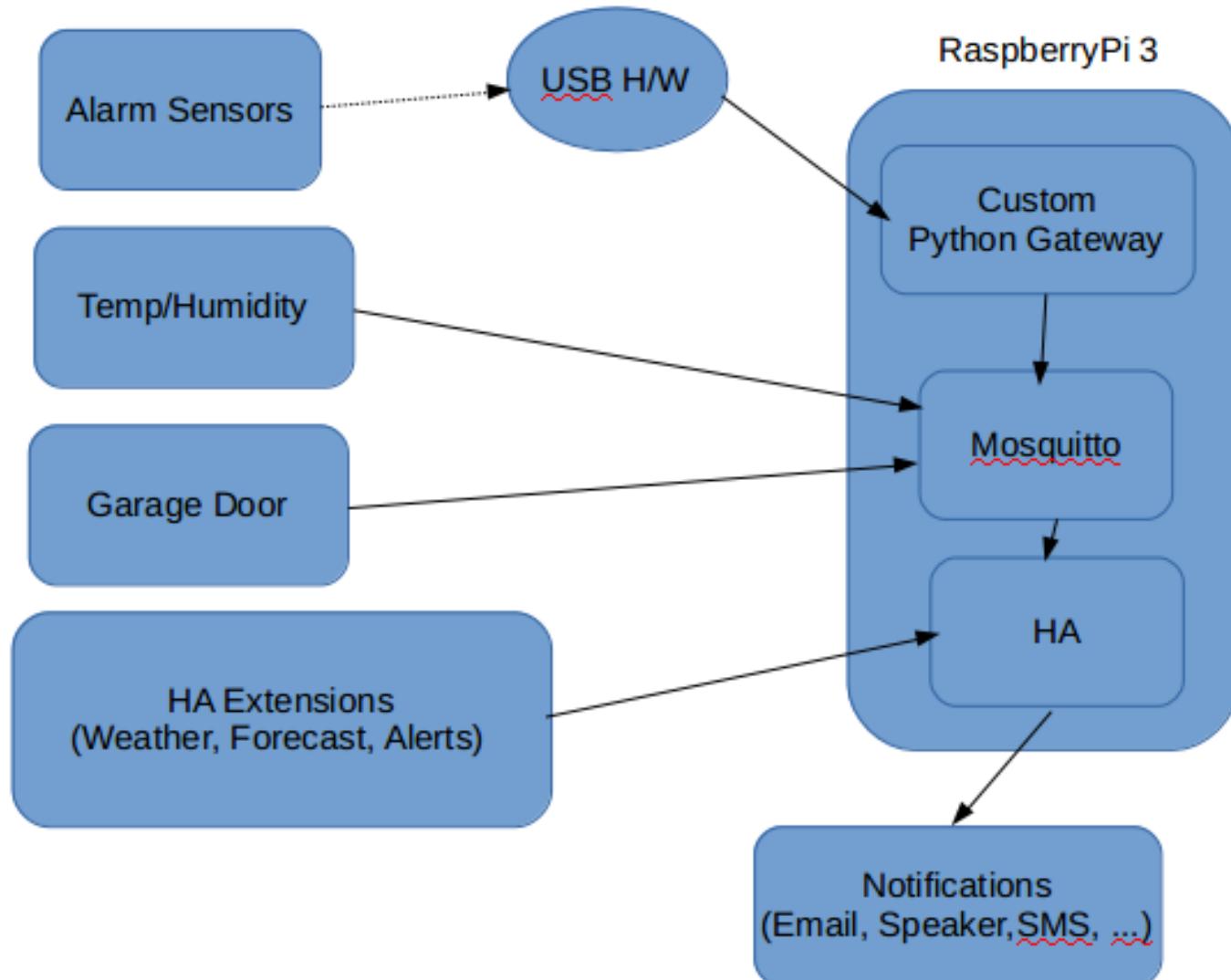


Five minute break ...





Home Assistant





Home Assistant



- Written in pure Python
- Installation methods
 - Hassio (custom RaspberryPi distro)
 - Python venv (inside Raspian)
- Interfaces to 1,100+ Devices/Services
- Many are ‘cloud’ services (which I avoid)





Home Assistant



Brief Tour ...





Goto Telescope



- OnStep firmware (GPL), C/C++, Arduino IDE
- Ported it to low cost STM32 ARM, by writing a Hardware Abstraction Layer
- MCU + ESP8266 WiFi + USB
- 2 X Stepper Motors + 2 X Stepper Drivers
- Tracks apparent motion of the sky
 - 15.04 arc seconds / second
- Commands via serial (USB or WiFi)
- Android App, desktop Planetariums





Goto Telescope



- Dec 2017: OnStep firmware worked on:
 - Arduino Mega 2560 (slow 16MHz, ~ \$12)
 - Teensy 3.2/3.5/3.6 (fast, ~ \$35)
- Summer 2018: Ported it to low cost STM32 ARM
 - By writing a Hardware Abstraction Layer
 - (72 MHz, ~ \$3)
- Designed a PCB, to make wiring easy
- MCU + PCB + odds and ends, only ~ \$35





OnStep

2bits

Demo





Other Stuff



- 3D Print a case for the OnStep controller
 - Design software?





Final Notes



- The Arduino ecosystem is not limited to Arduino hardware
- There are cheaper, smaller and more powerful options available
- China + eBay + patience = nice cheap toys
- Micropython: easier and more familiar





Conclusion



Questions?

Comments?

