



Intro to Arduino with C

Las Vegas C/C++ Meetup Group - Tuesday December 18th, 2018

Presented By: Ray Imber



Embedded Programming

- Embedded in a larger mechanical or Electrical system
- IOT
- Special Purpose : Microcontroller or FPGA
- Realtime



Languages Used in Embedded

- C and C++ is King (Often pure C or subset of C++)
- Assembly
- Ada / Spark
- Fourth
- VHDL / Verilog (Hardware Description Languages)
 - Describe physical circuits
- Special versions of Python and Javascript exist
 - Mostly used by hobbyists



Programming for Embedded Systems is Different

Like Stepping Back in time

- Much slower
 - ESP8266 is 80MHz
 - can overclock to 160MHz
- Single Threaded
- 32 bit processor is the norm
 - but 16 bit or 8 bit is also common
- Small memory
 - ESP8266 has 32Kib instruction memory & 80Kib user memory
- Direct access to hardware / “specialized” architectures





No Operating System

- Embedded Systems are **Special Purpose**
 - Usually only do one thing
- Embedded Systems often are **Real Time systems**
 - Application code must run as fast as possible
- Embedded systems must be **Power Efficient**
 - Less code to run means less battery used

Bugs crash the whole system! No OS to clean up after you!



No Standard Library

- The Standard Library is designed for desktop computers
 - SIMD assembly instructions don't exist on embedded
 - IO buffering doesn't make sense when you don't have a hard drive
- The Standard Library is big and embedded has very limited space
- Many library containers make large use of the heap and embedded has a very small heap
- Standard Library is designed to take advantage of desktop CPU assembly languages.
 - Embedded Systems have very small / simple assembly language

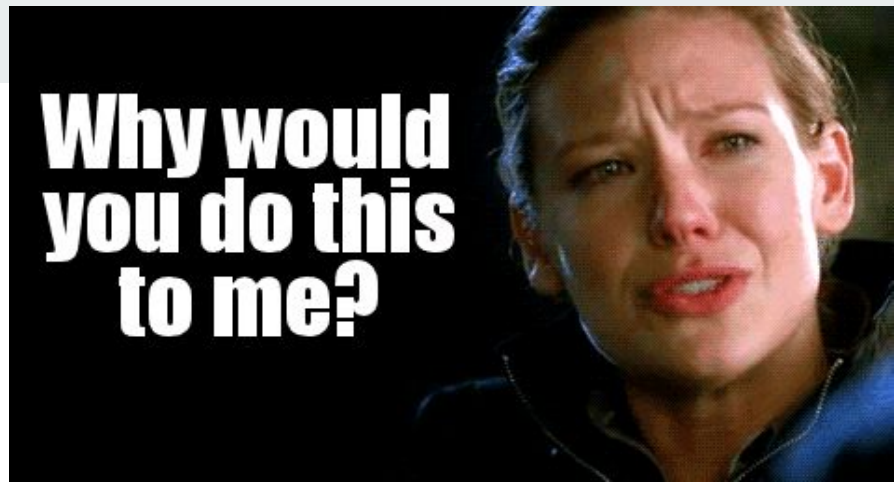


The Non Standard Library

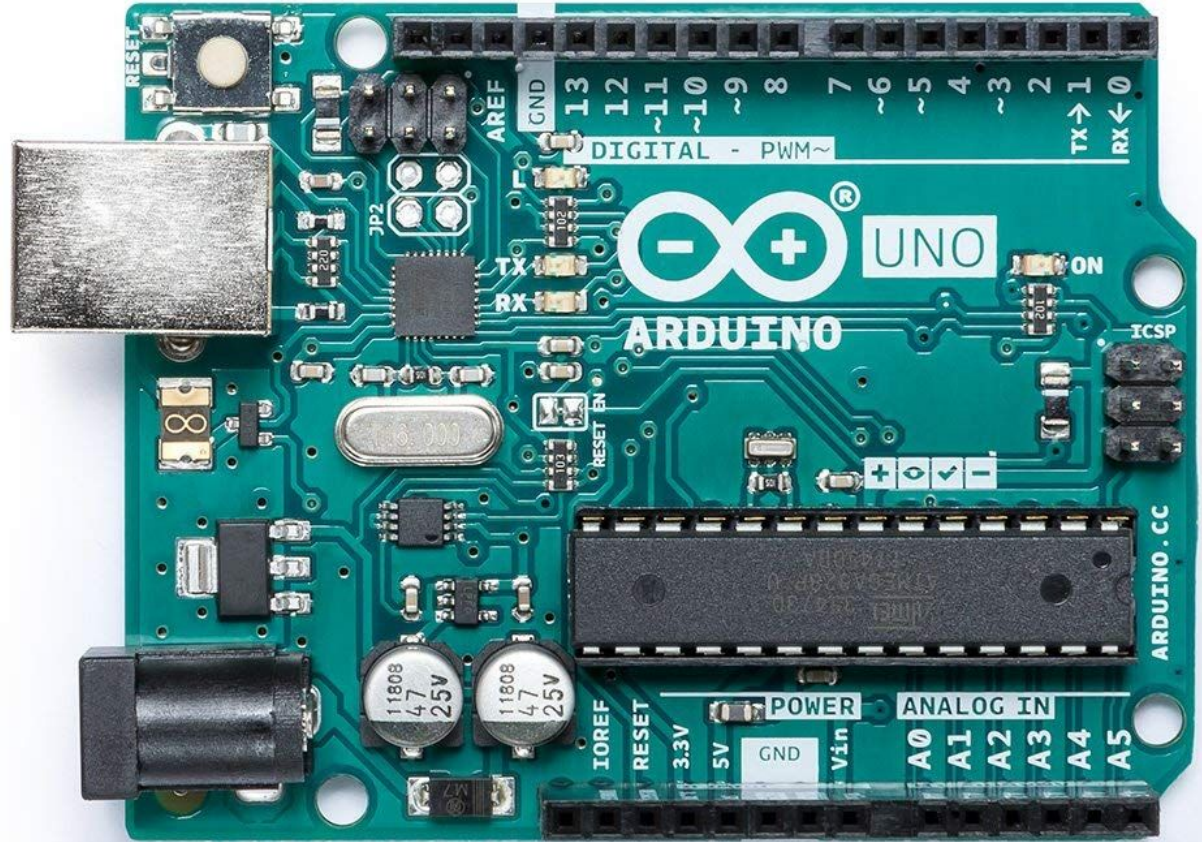
- Special Purpose libraries designed specifically for embedded systems
- RTOS : Real Time Operating System
 - Not a real operating system
 - A set of libraries to help write embedded code
 - Arduino SDK is a RTOS
- Embedded CPU manufacturers often provide libraries to go with their chips

WHY

- IOT is everywhere
- A feeling of complete control of the hardware
- Constraints make you more creative and more skilled!
- Teaches you how computers work on a more fundamental level
 - Principles are fundamentally the same for Desktop CPUs
 - Just more advanced / more layers of software to help you




Arduino





What is Arduino

- Not just one Chip
- A Company and a platform
 - DIY Kits
 - SDK
 - Accessories
- Popular for hobbyists because they are cheap and easy to program



Arduino Uno

<https://store.arduino.cc/usa/arduino-uno-rev3>



ESP8266

- ~ 5.00\$ a piece (includes chip + development board)
- Originally Designed as a Wifi controller
- Community found you could use it as a general purpose microcontroller
- More powerful than the Arduino Uno!
- Has been ported to work with the Arduino SDK and IDE
- <https://en.wikipedia.org/wiki/ESP8266>
- <https://www.amazon.com/s/field-keywords=esp8266>



Hello World Arduino



Arduino Studio

- Download and Install Arduino Studio
 - <https://www.arduino.cc/en/Main/Software>
- Download and install the esp8266 sdk
 - <https://arduino-esp8266.readthedocs.io/en/2.5.0-beta1/installing.html#boards-manager>



Notes for Mac

For mac you may need to install additional USB Serial drivers:

- <https://www.silabs.com/products/development-tools/software/usb-to-uart-bridge-vcp-drivers>
- <https://learn.sparkfun.com/tutorials/how-to-install-ftdi-drivers/all#yosemite>



Hello World Arduino

<https://bit.ly/2ScNPjv>



Anatomy of Arduino Code

- `No int main()!`
- This is an Arduino SDK thing!
- `Void setup()`
 - Initialization code
 - Sets up hardware (remember no OS)
- `Void loop()`
 - Main code must run in a loop
 - your code is the only code; there is nothing to exit to!
 - The closest you can do is to just halt the processor.



Flashing the ESP Board

1. Press and hold RST button.
2. Press and HOLD FLASH button while RST is in pressed condition.
3. Release RST while FLASH is in pressed condition or on some boards hold it during program upload.
4. Release FLASH after releasing RST



Notes about Arduino IDE

- IDE Written in Java
- Compiler is modified gcc for embedded systems (based on gcc-5.4.0)
 - <https://github.com/arduino/toolchain-avr>
- Does support C++ 11 (mostly)
 - Template meta-programming - Yes
 - Exceptions - No



A Packet Sniffer

<https://bit.ly/2A2sHoY>

Courtesy of @Kalanda on Github

Legal Note* *This code is only for educational purposes only! I or anyone associated with the C++ meetup group is not liable for anything you do with this code or these techniques. Please, check which country's laws applies to you before use this code.*



Anatomy of the Packet Sniffer

- `#include <user_interface.h>`: C api from the manufacturer
- `Consts` and `Structs` representing the wifi packets
- `showMetadata` : helper function that does all the work
- `Sniffer_callback`: wifi hardware calls this function
 - This is a hardware interrupt handler
- Other Helper functions
- `setup()` : registers the interrupt handler and sets up the hardware
- `loop()` : empty loop to keep the arduino sdk happy.
 - QUIZ! Why `delay()` and not just an infinite loop?



Bonus Challenge

Make the Arduino blink 3 times when it detects your phone's mac address



References

- <http://www.idogendel.com/en/archives/19#comment-2809>
- <https://electrosome.com/esp8266-arduino-programming-led-blink/>
- <https://arduino-esp8266.readthedocs.io/en/2.5.0-beta1/index.html>