Lecture 27

Wrapup

Agenda for Last MEAM510 Lecture

- Beyond Copy/Past Development
- Recap of lecture topics
- Short Quiz
- Future resources for more Mechatronics

Reminders:

- Don't forget to do the scope activity.
  - It may be particularly useful to borrow a scope if you are using the oscillosorta ESP32 for your robot.
- Final Report Documentation
  - Document progress.
  - Everytime you get something working, take a short video (so you can prove things were working – in case it breaks later).
  - Before dropping robot off at GMlab take some nice photos.
  - Plan to get practice on the field before grading. You will want to practice at least twice.

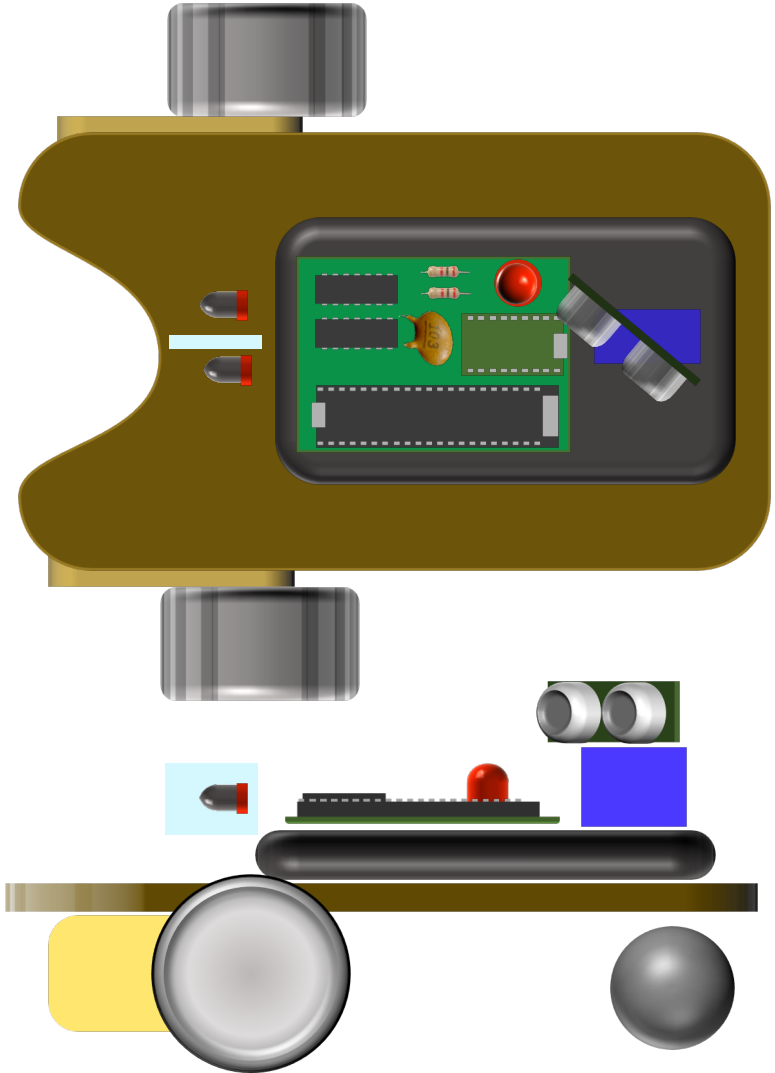
01

Class Architecture

Beyond copy-paste development

Class Structure

- Goal is final project
- Learn pieces
- Learn why pieces work
- Integrate in stages
  - Lab 1: Micro
  - Lab 2: Beacon
  - Lab 3: Waldo
  - Lab 4: Mobility
- Copy/Paste Integrate

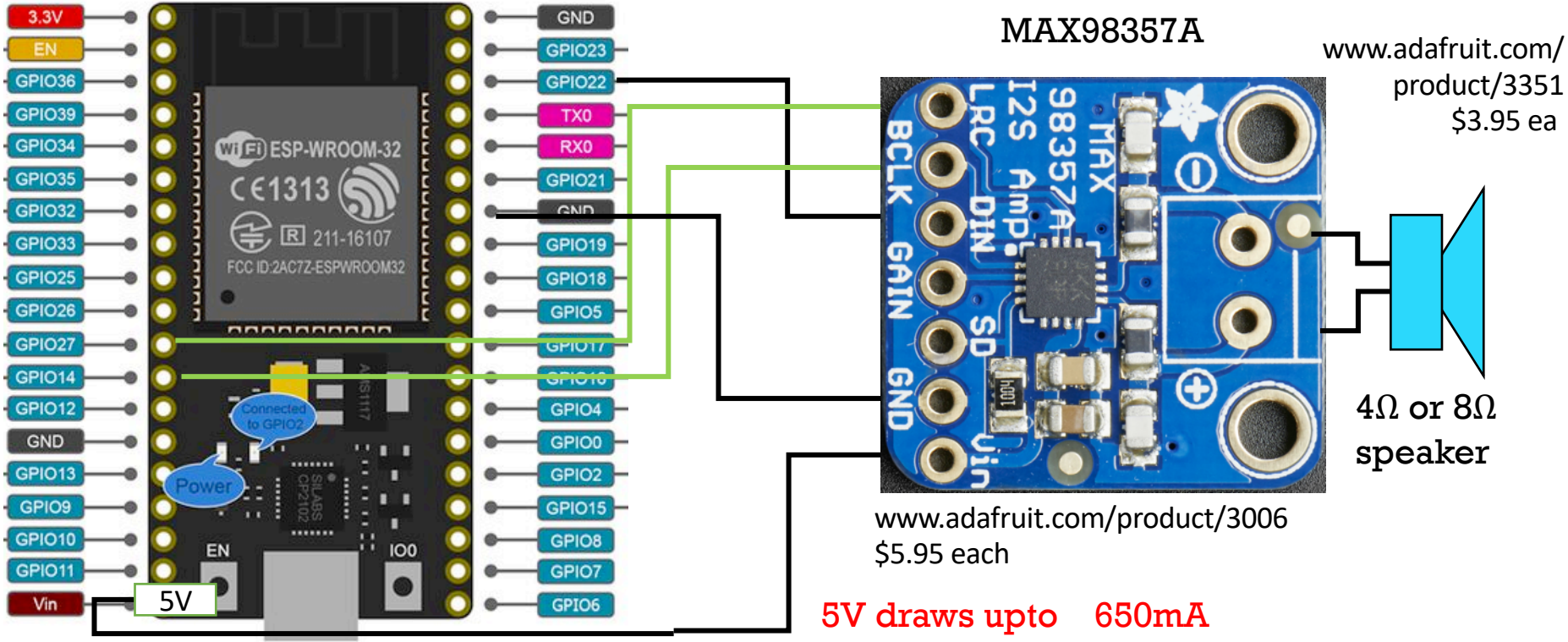


Navigating the Arduino Free Software World

1. You find some site that has software – download – test
2. Try to modify so it works for your case.
3. Find library that code relies on.

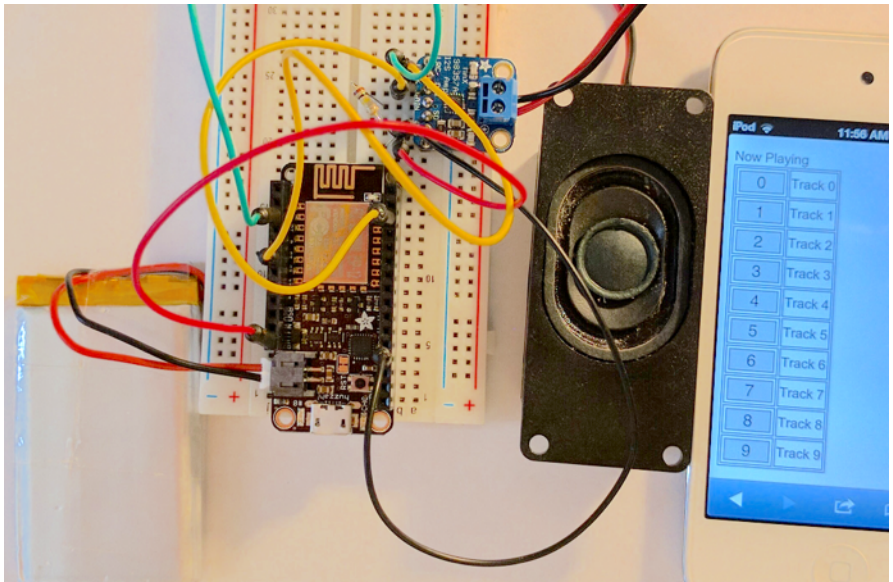
   http://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use
4. Search for Library source code. (typically github)
5. Debug the mistakes from public domain software.
6. Learn and use new system.

I2S on Arduino on ESP32



Arduino I2S Example Scenario (from 2018)

- Google I2S ESP8266
  - https://github.com/bbx10/SFX-I2S-web-trigger
  - I2S from SPIFFS file plus web interface
- Set up and run
  - Works!
- Change to fit
  - Remove web stuff
  - Upload different sound files
- Result
  - Crashes sometimes
  - Doesn't play some sound files



Arduino I2S Example Scenario: Bug #1

- Problem: crashes
  - Play same 3 files over and over (in loop())
  - After 10 plays it crashes. (every time).
- Google I2S library
  - github.com/esp8266/Arduino/blob/master/cores/esp8266/core_esp8266_i2s.c
- Examination of I2S C code shows `i2s.begin()` uses `malloc()` and `i2s.end()` uses `free()`.
- The Arduino code calls begin() and end() for every song file.
  - malloc() and free() tend to fragment memory and overtime things crash.
  - Solution: call begin() once in setup() instead.

For embedded code – avoid repeated calls to malloc() and free(). Usually, you know how much space you need. Pre-allocate in a global.

Arduino I2S Example Scenario: Bug #2

- Problem: sometimes sound crackles or drops
  - Especially if moving wires.
- The I2S runs at high frequencies.
- The sound amp takes high currents (~700mA)
- Push on individual pins to see which is the problem.

Q1: What could be the issue?

- Solution: use better wire connections. Resolder pins.

Sometimes hardware is the problem – but usually suspect software first.

Arduino I2S Example Scenario: Bug #3

- Problem: crashes
  - Simplifying cleaning up code…
  - Commenting out Serial.println() causes a crash.
- Must be a timing issue.
- Program writes bytes (e.g. a sound file) to a DMA (direct memory access) subsystem in big chunks. DMA writes those bytes one at a time to I2S port without processor babysitting.
- Problem DMA buffer was not clearing in time. Didn't think about wav_loop() calls happening quickly in succession.
- Solution: use library i2s_is_full() routine instead of own flags.

If removing println() makes something crash – suspect a timing issue.

Summary

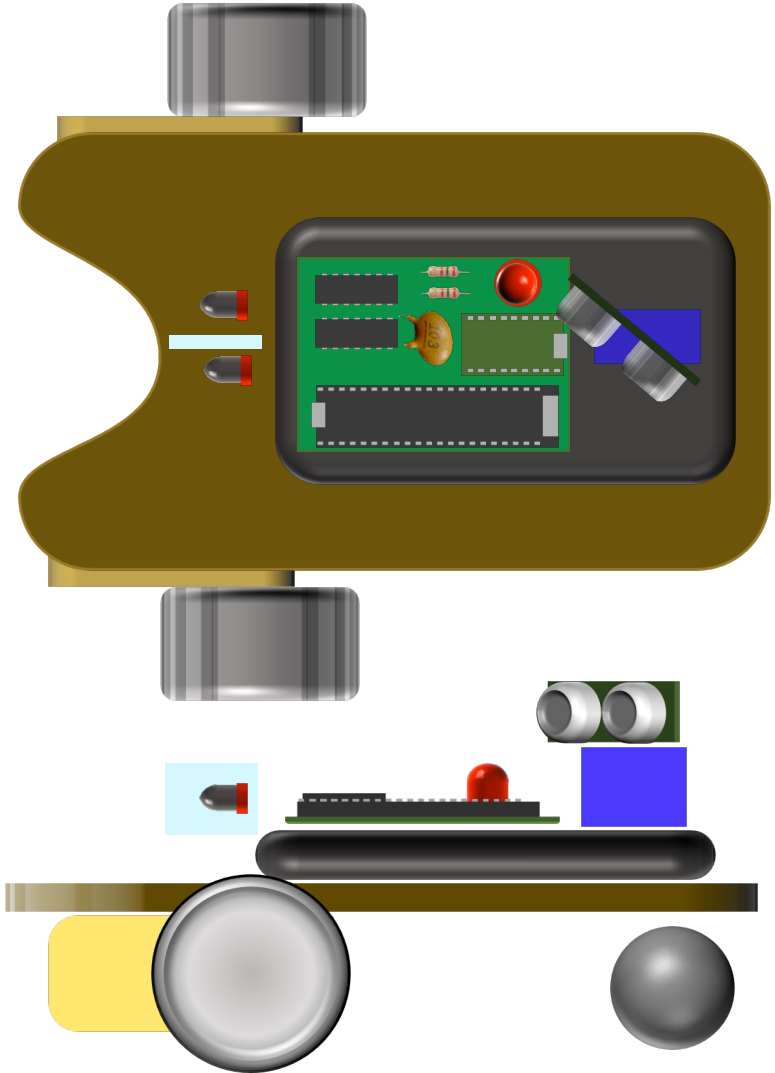
- Sample Arduino code is everywhere. Quality is random.
- If you see random crashes for repeated code (without interrupts) suspect a memory leak or fragmentation. Look for overusing malloc() and free().
- If removing println() causes crashes, suspect a timing issue.
- If you have flakey hardware, push on wires to find suspect connections.

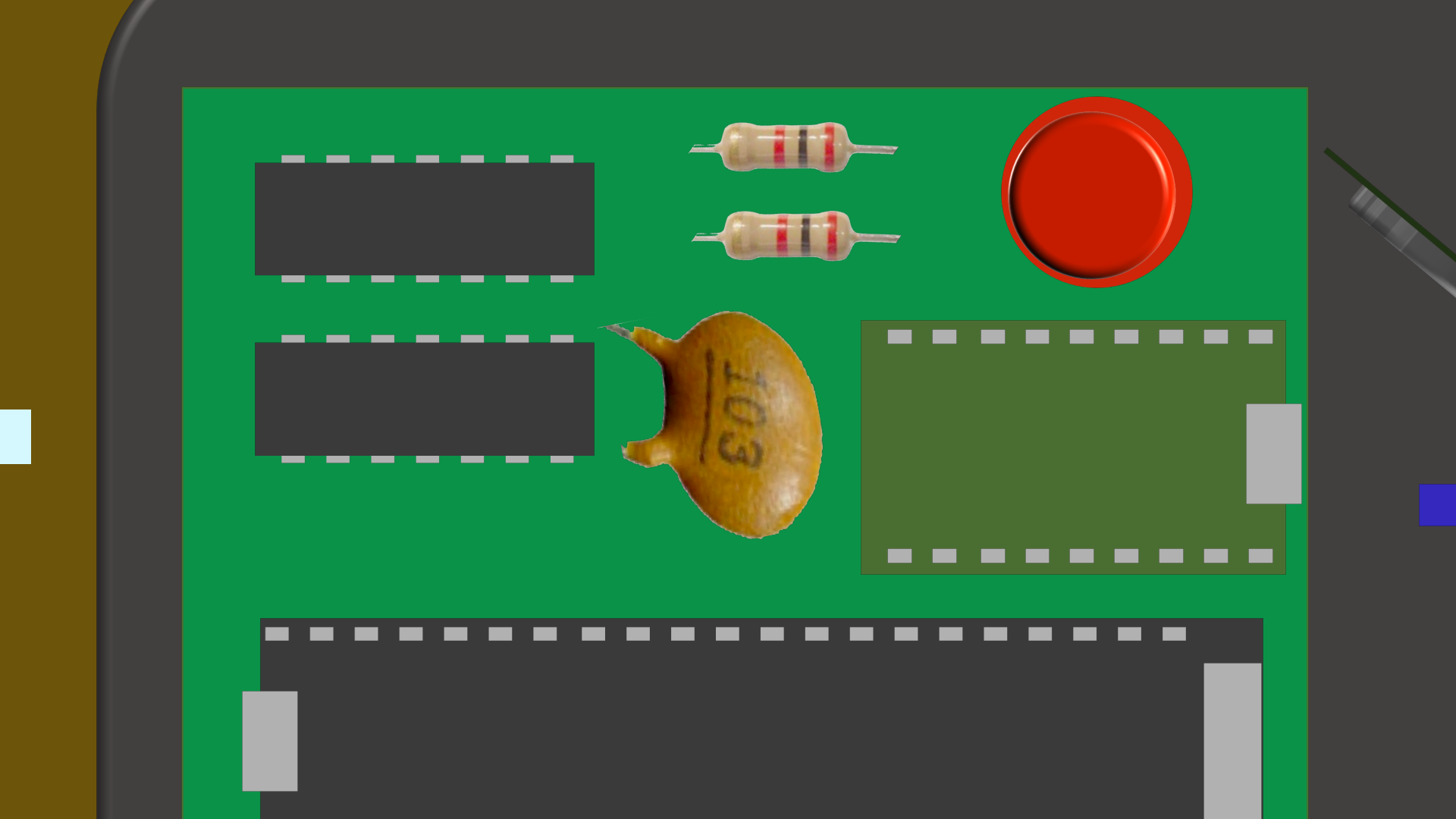
02

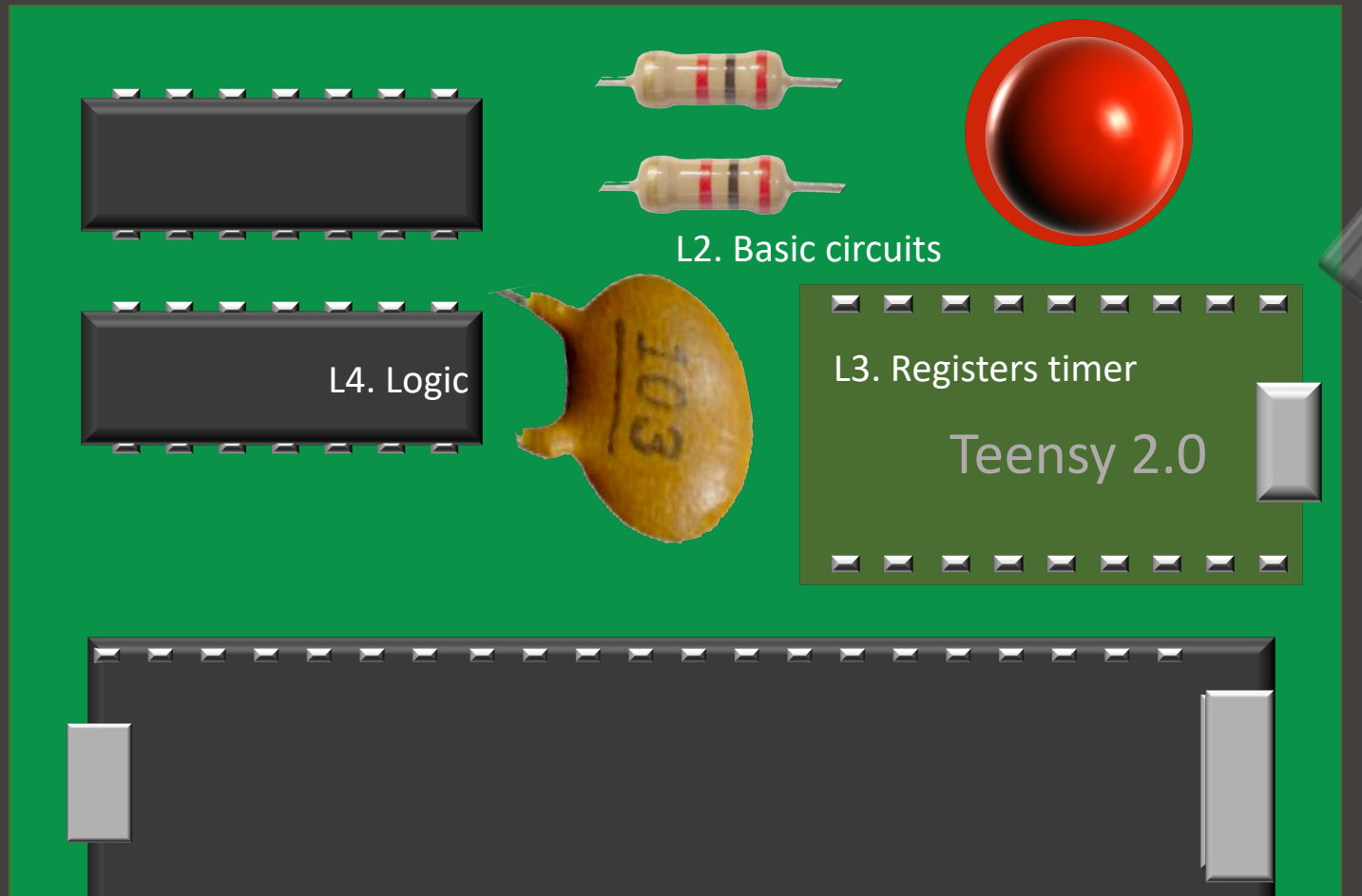
Past Lecture Recap

Class Structure

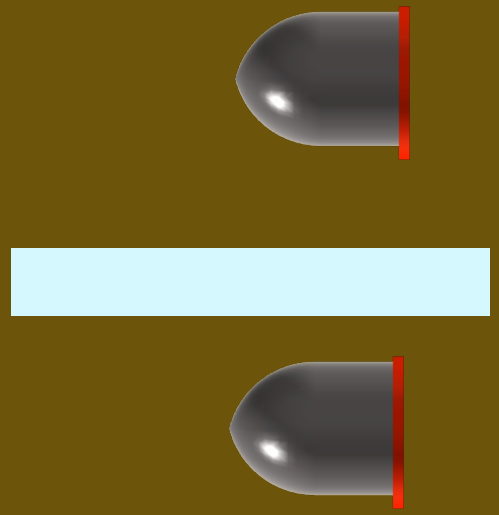
- Goal is final project
- Learn pieces
- Learn why pieces work
- Integrate in stages
  - Lab 1: Micro
  - Lab 2: Beacon
  - Lab 3: Waldo
  - Lab 4: Mobility

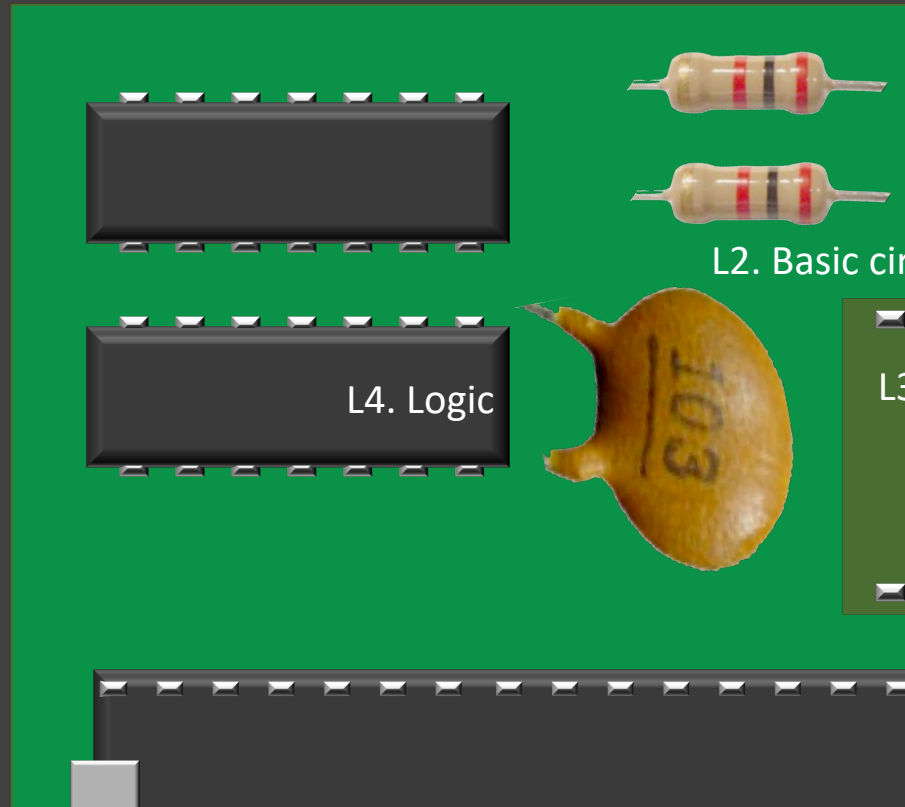



103


L2. Basic circuits
L4. Logic
L3. Registers timer
Teensy 2.0

L5. Timers, Analog filters
L6. Photo sensor opamps
L7. Comparators
L20. Beacon Tracking





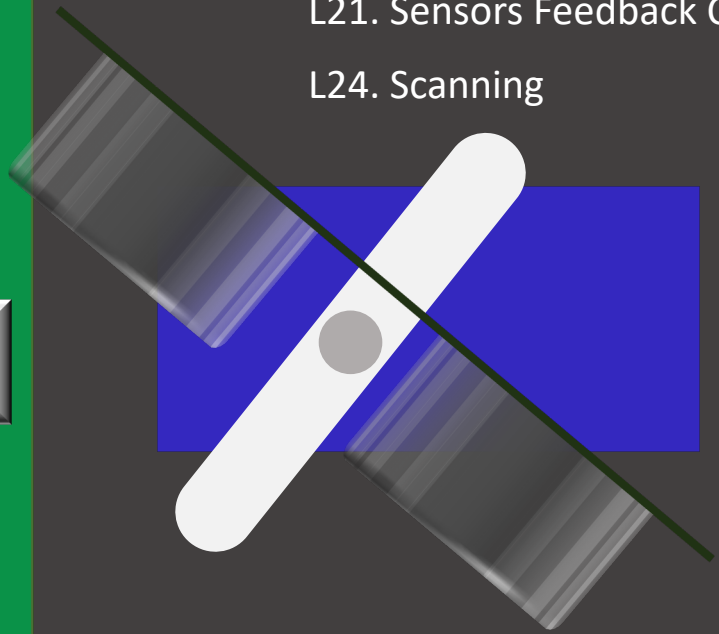
L2. Basic circuits

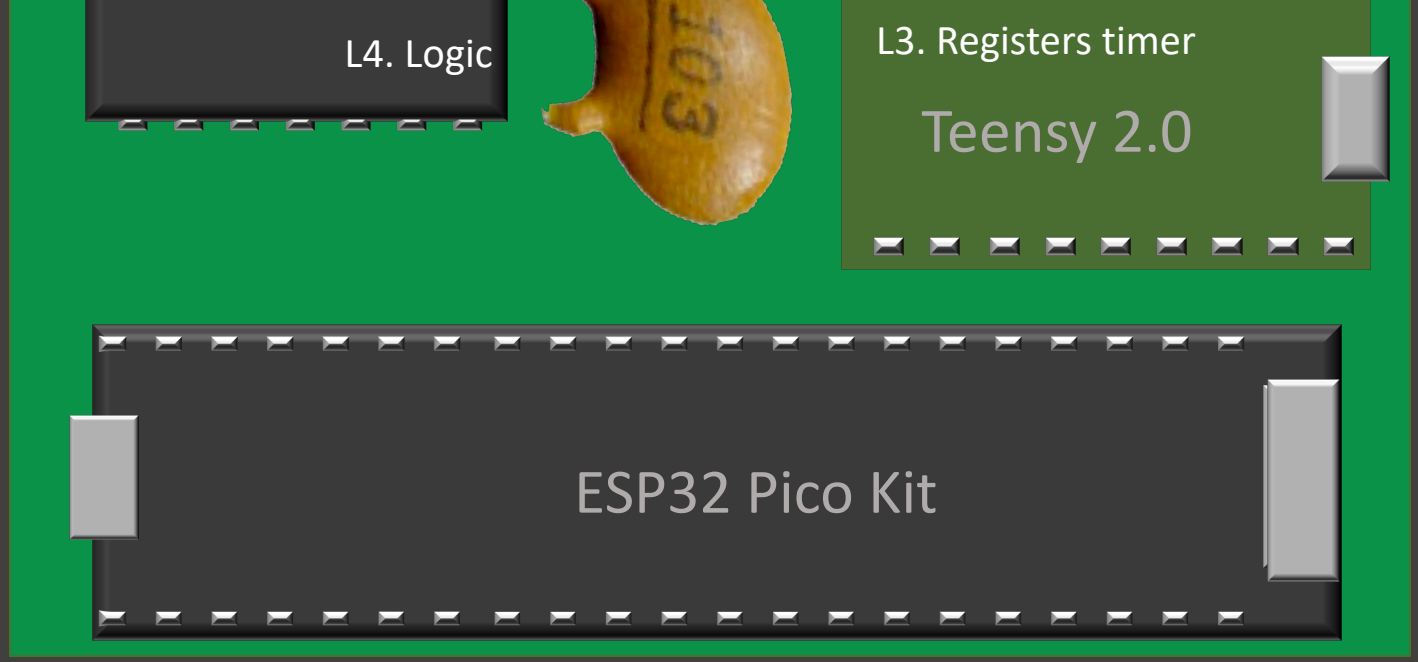
L3. Registers timer

L10. RC servos

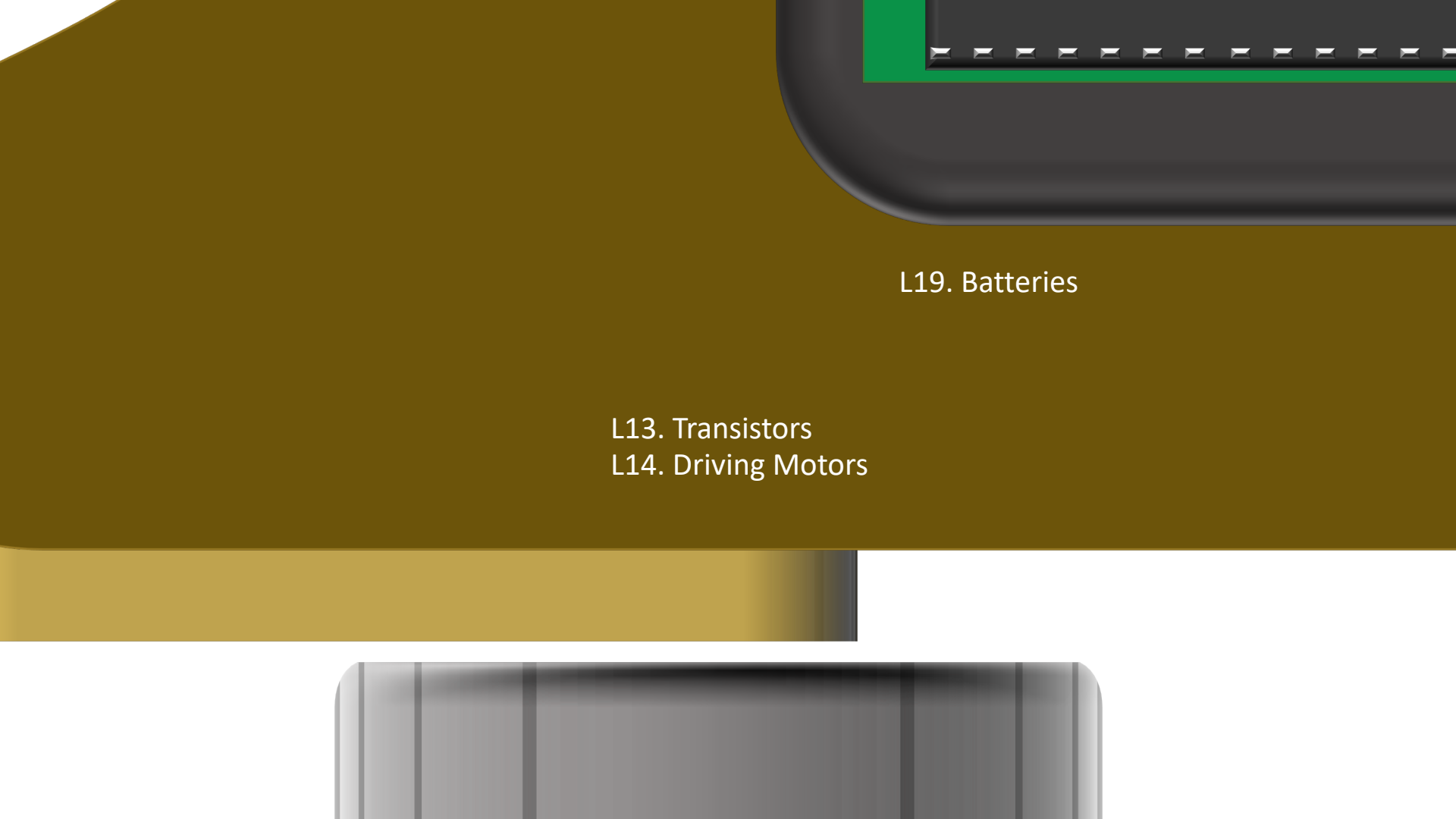
L21. Sensors Feedback Control

L24. Scanning

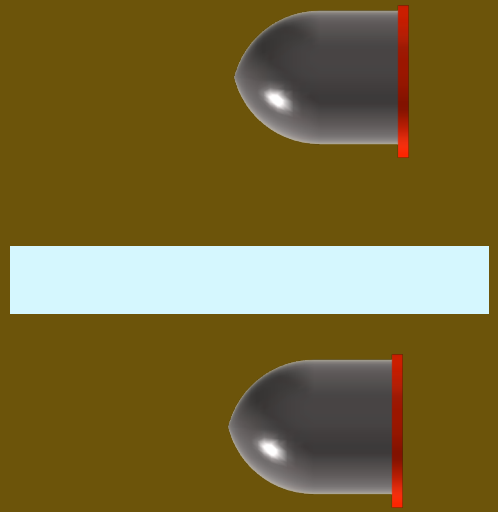




L15. ESP32
L16. WiFi
L17. HTML internet
L18. ESP interrupts


L19. Batteries
L13. Transistors
L14. Driving Motors

L3. Timers, Analog filters
L6. Photo sensor opamps
L7. Comparators
L20. Beacon Tracking



L8. ADC and Digital filters
L9. Waldo
L11. Pointers, Event-based programs
L12. Interrupts
L22. Wired Comms
L23. SPI UART I2C
L25. Noise and debugging

03

Review Quiz

Slides omitted

04

More Mechatronics

Some Online Resources

Learning C

- http://www.learn-c.org/

OpAmp Tutorial

- http://www.electronics-tutorials.ws/opamp/opamp_1.html

Advanced OpAmps - Book edited by Walt Jung available online

- http://www.analog.com/en/education/education-library/op-amp-applications-handbook.html

ESP32 ESP-IDF – not Arduino, but reasonable

- https://docs.espressif.com/projects/esp-idf/en/latest/esp32/

Noise

- Noise Reduction Techniques by Henry Ott

Other Resources For Electronics

- Digikey.com
  - Quick highlight of important parameters for classes of devices

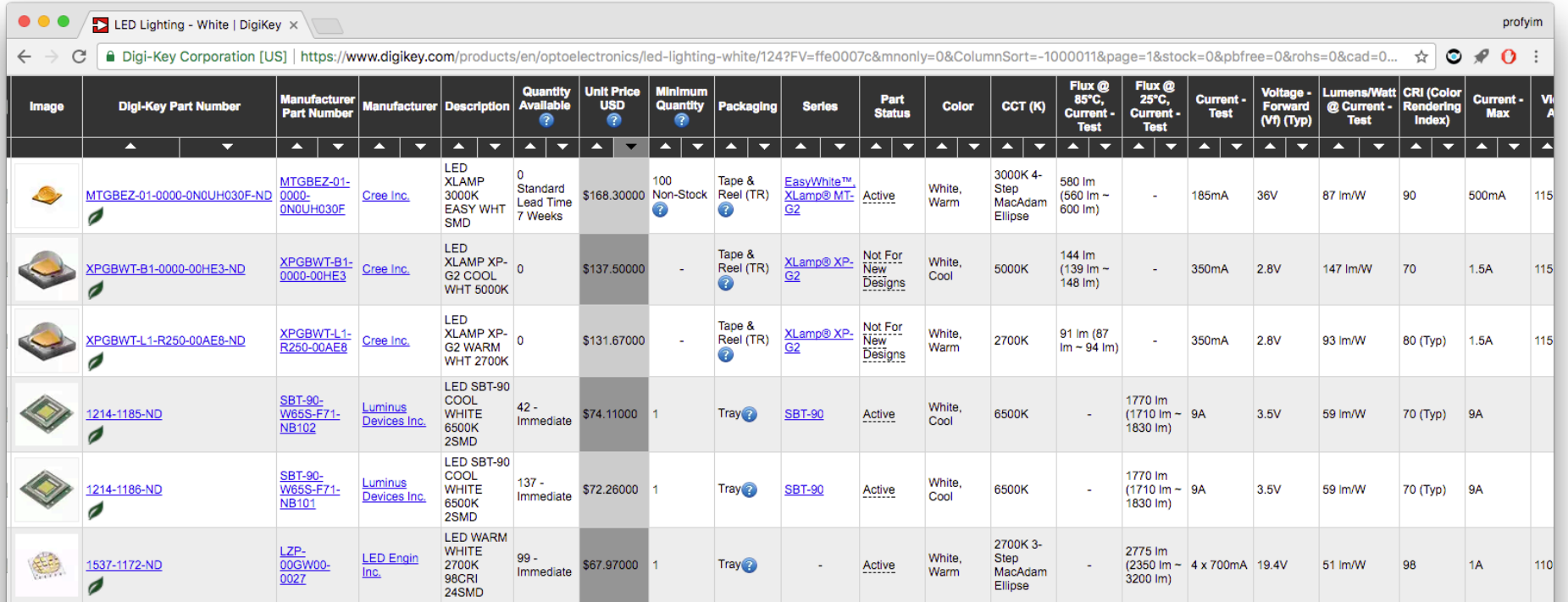


Image	Digi-Key Part Number	Manufacturer Part Number	Manufacturer	Description	Quantity Available	Unit Price USD	Minimum Quantity	Packaging	Series	Part Status	Color	CCT (K)	Flux @ 85°C, Current - Test	Flux @ 25°C, Current - Test	Current - Test	Voltage - Forward (Vf) (Typ)	Lumens/Watt @ Current - Test	CRI (Color Rendering Index)	Current - Max	Vi A
	MTGBEZ-01-0000-0N0UH030F-ND	MTGBEZ-01-0000-0N0UH030F	Cree Inc.	LED XLAMP 3000K EASY WHT SMD	0 Standard Lead Time 7 Weeks	$168.30000	100 Non-Stock	Tape & Reel (TR)	EasyWhite™ XLamp® MT-G2	Active	White, Warm	3000K 4-Step MacAdam Ellipse	580 lm (560 lm ~ 600 lm)	-	185mA	36V	87 lm/W	90	500mA	115
	XPGBWT-B1-0000-00HE3-ND	XPGBWT-B1-0000-00HE3	Cree Inc.	LED XLAMP XP-G2 COOL WHT 5000K	0	$137.50000	-	Tape & Reel (TR)	XLamp® XP-G2	Not For New Designs	White, Cool	5000K	144 lm (139 lm ~ 148 lm)	-	350mA	2.8V	147 lm/W	70	1.5A	115
	XPGBWT-L1-R250-00AE8-ND	XPGBWT-L1-R250-00AE8	Cree Inc.	LED XLAMP XP-G2 WARM WHT 2700K	0	$131.67000	-	Tape & Reel (TR)	XLamp® XP-G2	Not For New Designs	White, Warm	2700K	91 lm (87 lm ~ 94 lm)	-	350mA	2.8V	93 lm/W	80 (Typ)	1.5A	115
	1214-1185-ND	SBT-90-W65S-F71-NB102	Luminus Devices Inc.	LED SBT-90 COOL WHITE 6500K 2SMD	42 - Immediate	$74.11000	1	Tray	SBT-90	Active	White, Cool	6500K	-	1770 lm (1710 lm ~ 1830 lm)	9A	3.5V	59 lm/W	70 (Typ)	9A	
	1214-1186-ND	SBT-90-W65S-F71-NB101	Luminus Devices Inc.	LED SBT-90 COOL WHITE 6500K 2SMD	137 - Immediate	$72.26000	1	Tray	SBT-90	Active	White, Cool	6500K	-	1770 lm (1710 lm ~ 1830 lm)	9A	3.5V	59 lm/W	70 (Typ)	9A	
	1537-1172-ND	LZP-00GW00-0027	LED Engin Inc.	LED WARM WHITE 2700K 98CRI 24SMD	99 - Immediate	$67.97000	1	Tray	-	Active	White, Warm	2700K 3-Step MacAdam Ellipse	-	2775 lm (2350 lm ~ 3200 lm)	4 x 700mA	19.4V	51 lm/W	98	1A	110

Reading Datasheets

- http://www.egr.msu.edu/classes/ece480/capstone/read_datasheet.pdf

- www.sparkfun.com/tutorials/223

There will always be a date. Datasheets do change, especially if Preliminary or Advance. Check the date!

National Semiconductor

February 2000

Look up here to see if the datasheet is Advance Information or Preliminary.

LM555 Timer

Features tell you general characteristics--always check the Electrical Characteristics for conditions and exceptions.

General Description

The LM555 is a highly stable device for generating accurate time delays or oscillation. Additional terminals are provided for triggering or resetting if desired. In the time delay mode of operation, the time is precisely controlled by one external resistor and capacitor. For astable operation as an oscillator, the free running frequency and duty cycle are accurately controlled with two external resistors and one capacitor. The circuit may be triggered and reset on falling waveforms, and the output circuit can source or sink up to 200mA or drive TTL circuits.

Sometimes the General Description will tell you about a feature or usage not mentioned anywhere else! For example, you might need to hold a specific pin low for some operation.

Features

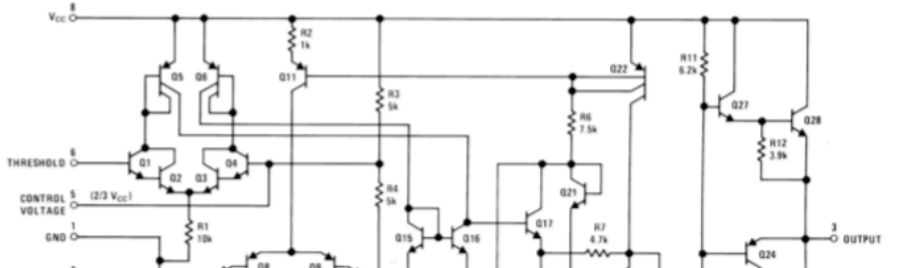
- Direct replacement for SE555/NE555
- Timing from microseconds through hours
- Operates in both astable and monostable modes
- Adjustable duty cycle
- Output can source or sink 200 mA
- Output and supply TTL compatible
- Temperature stability better than 0.005% per °C
- Normally on and normally off output
- Available in 8-pin MSOP package

Applications

- Precision timing
- Pulse generation
- Sequential timing
- Time delay generation
- Pulse width modulation
- Pulse position modulation
- Linear ramp generator

Application suggestions can often tell you quickly if this device is in the ballpark for what you want to do, but these lists are often very general.

Schematic Diagram



Some nice ESP32 youtubers

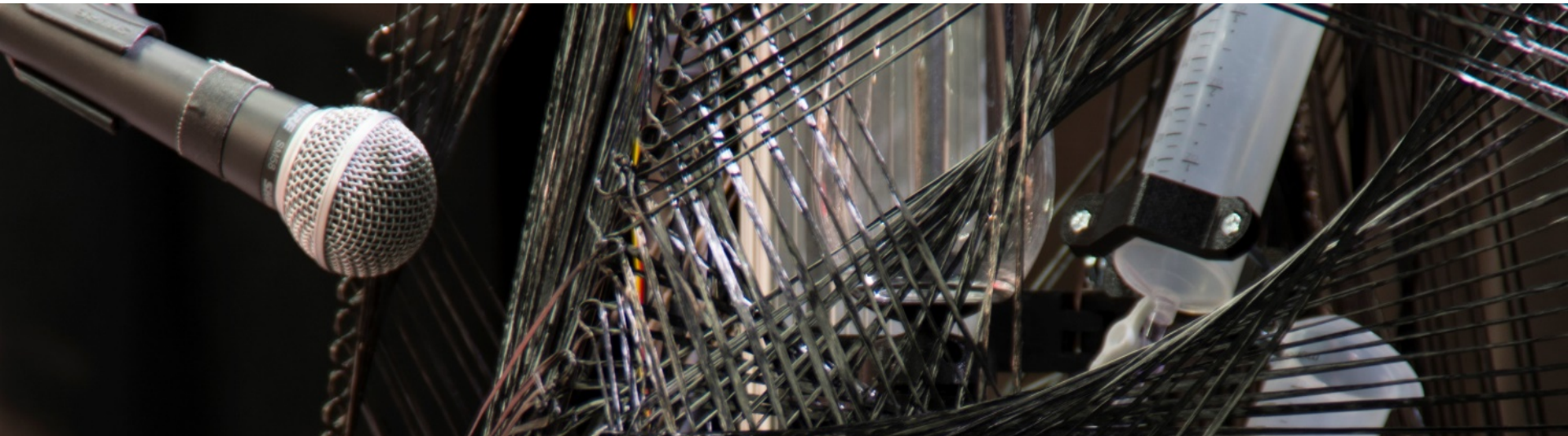
Indepth teaching descriptions (including github links to code)

- Andreas Spiess – tools RF WiFi remote comparisons
  - https://www.youtube.com/channel/UCu7_D0o48KbfhpEohoP7YSQ
- Atomic14 – sensors audio laser projects
  - https://www.youtube.com/channel/UC4Otk-uDioJN0tg6s1QO9lw
- Bitluni – wacky projects – CNC camera 3D displays
  - https://www.youtube.com/user/bitlunislab

https://youtu.be/T_n8PtMMLiQ?t=351

Future Classes

- IPD516 Advanced Mechatronics in Reactive Spaces
  - Maybe Offered Spring 2022 (1 years from now)
  - Performance art combined with architecture
- I will be looking for TA's for the Fall 2021 Semester!

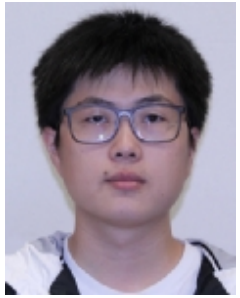


Things still to do (reminder)

- Oscilloscope Activity for ½ the class
- Final Project
- Final Project Report
- Feedback Survey on Piazza (important to me)
- Think about TA in the Fall

Thanks to the TA's

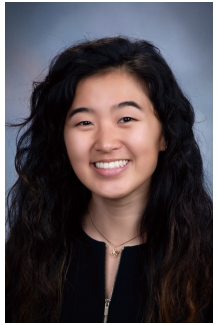











Thanks!

- For bearing with the teaching staff during this on-line hybrid experiment.
- For paying attention and working on learning with us.
- For making the best of challenging times.