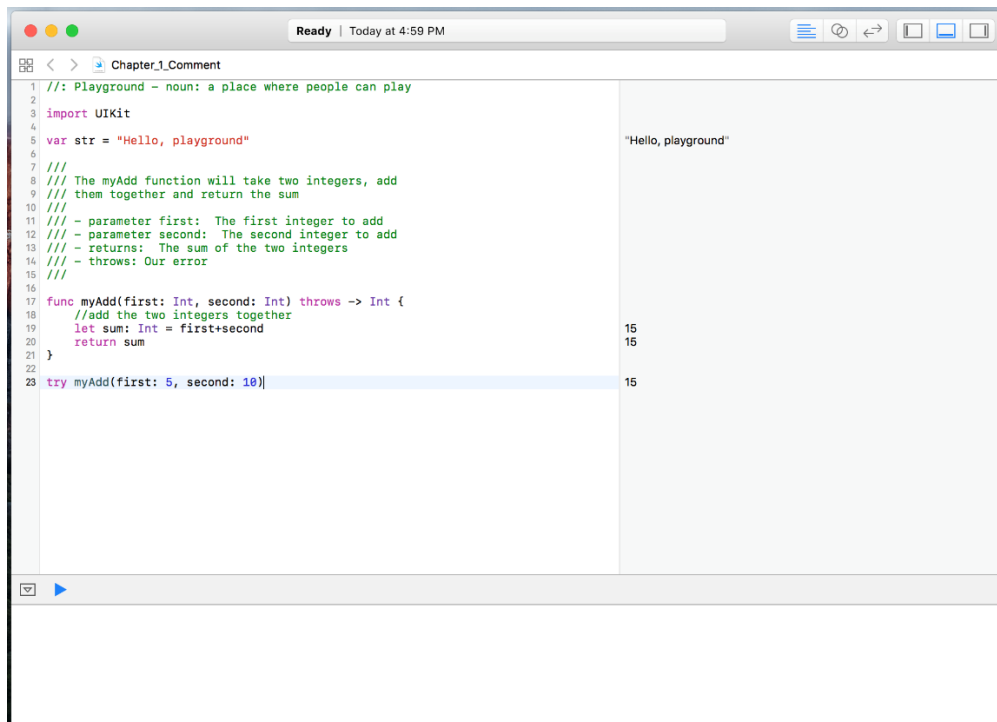


Chapter 1: Taking the First Steps with Swift



The screenshot shows an Xcode playground window titled "Chapter_1_Comment". The window is split into two panes. The left pane contains Swift code, and the right pane shows the output of the code. The code includes a comment, an import statement, a string variable, a function definition with documentation, and a function call. The output shows the string value and the result of the function call.

```
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
7 ///
8 /// The myAdd function will take two integers, add
9 /// them together and return the sum
10 ///
11 /// - parameter first: The first integer to add
12 /// - parameter second: The second integer to add
13 /// - returns: The sum of the two integers
14 /// - throws: Our error
15 ///
16
17 func myAdd(first: Int, second: Int) throws -> Int {
18     //add the two integers together
19     let sum: Int = first+second
20     return sum
21 }
22
23 try myAdd(first: 5, second: 10)
```

Output:

```
"Hello, playground"
15
15
15
```

```
6
7 ///
8 /// The myAdd function will take two integers, add
9 /// them together and return the sum
10 ///
11 /// - parameter first: The first integer to add
12 /// - parameter second: The second integer to add
13 /// - returns: The sum of the two integers
14 /// - throws: Our error
15 ///
16
17 func myAdd(first: Int, second: Int) throws -> Int {
18     //add the two integers together
```

Declaration `func myAdd(first: Int, second: Int) throws -> Int`

Description The myAdd function will take two integers, add them together and return the sum

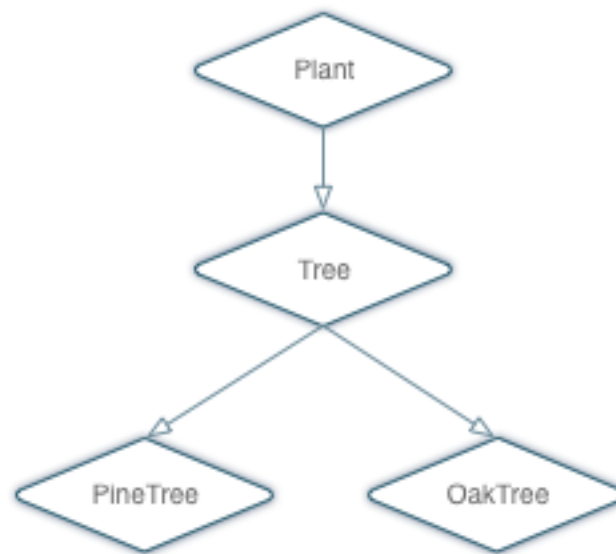
Parameters `first` The first integer to add
 `second` The second integer to add

Throws Our error

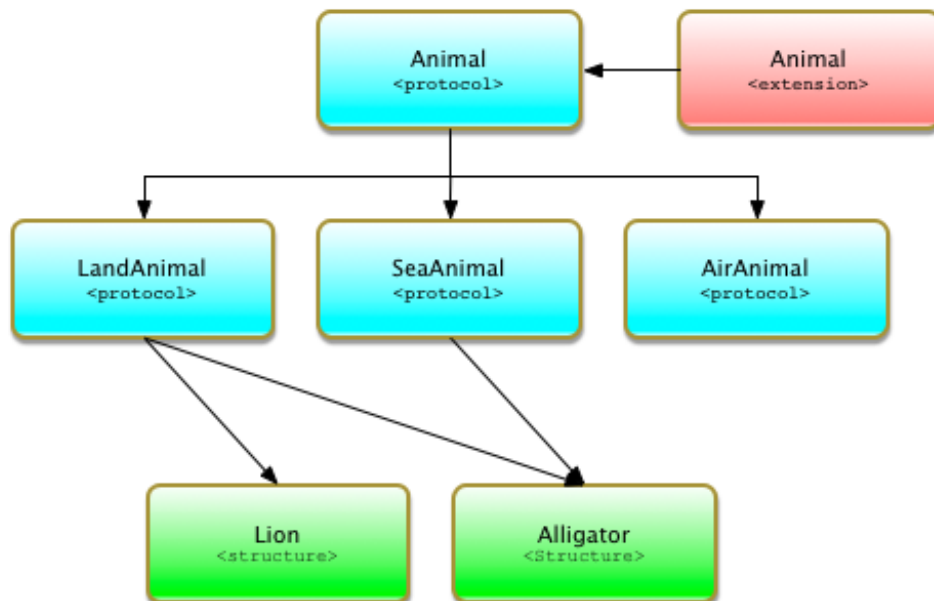
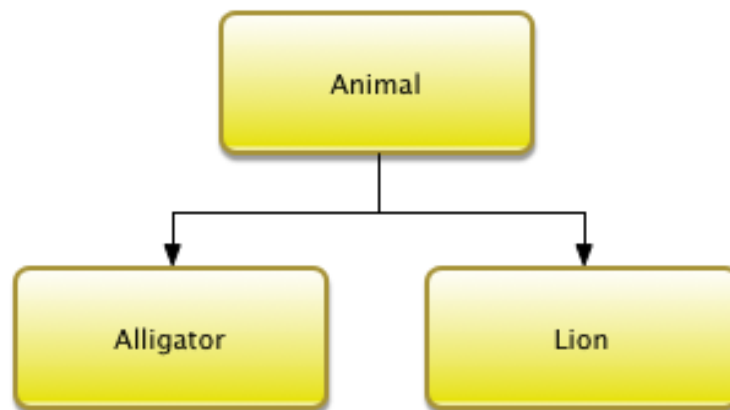
Returns The sum of the two integers

Declared In [Chapter_1_Comment.playground](#)

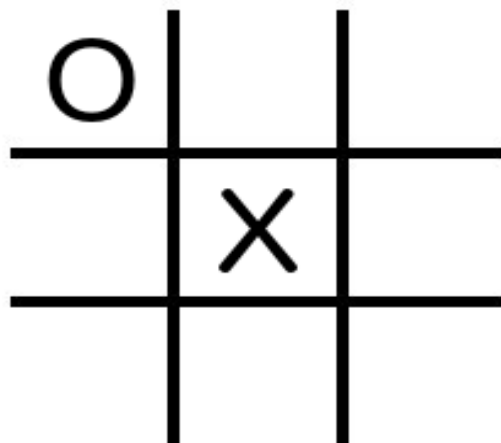
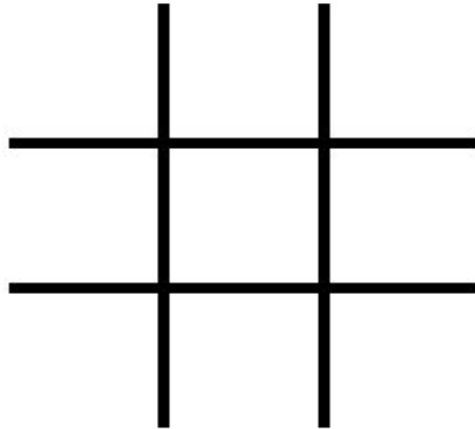
Chapter 5: Classes and Structures



Chapter 7: Protocol-Oriented Design



Chapter 9: Custom Subscripting



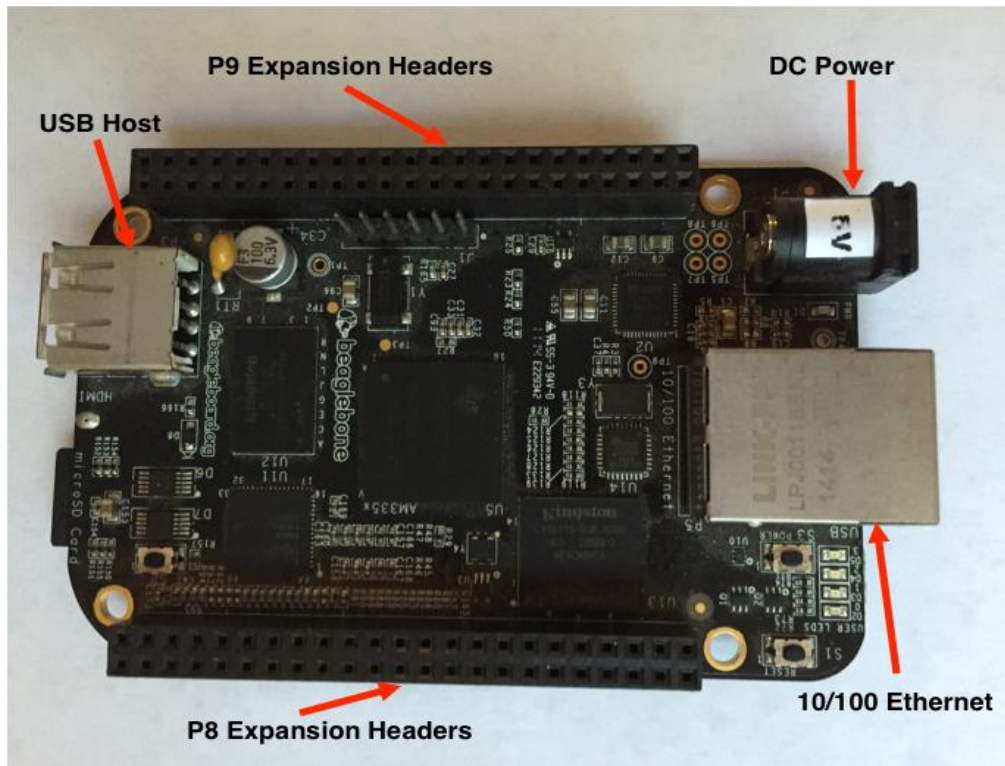
Chapter 13: Using C Libraries with Swift

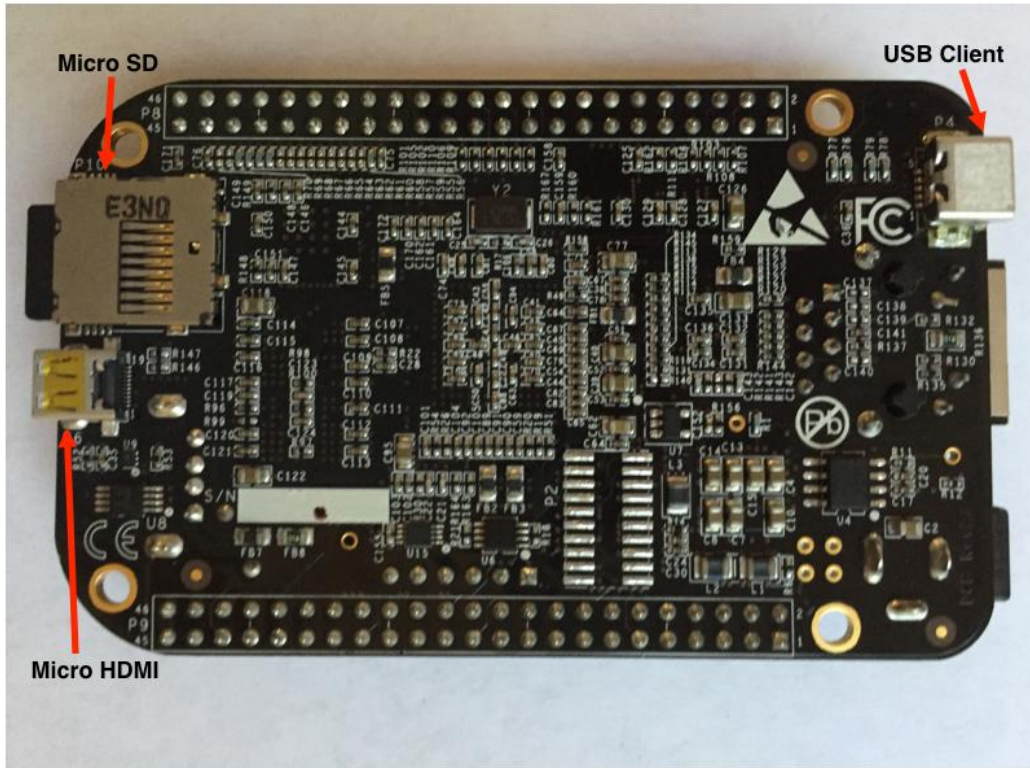
```
hoffmanjon@hoffmanjon-VirtualBox: ~/Dropbox/Books/Mastering Swift 3/Linux/Code/Chap
PCAP_FINDALLDEVS(3PCAP) PCAP_FINDALLDEVS(3PCAP)
NAME
    pcap_findalldevs, pcap_freealldevs - get a list of capture devices, and
    free that list
SYNOPSIS
    #include <pcap/pcap.h>

    char errbuf[PCAP_ERRBUF_SIZE];

    int pcap_findalldevs(pcap_if_t **alldevsp, char *errbuf);
    void pcap_freealldevs(pcap_if_t *alldevs);
DESCRIPTION
    pcap_findalldevs() constructs a list of network devices that can be
    opened with pcap_create() and pcap_activate() or with pcap_open_live().
    (Note that there may be network devices that cannot be opened by the
    process calling pcap_findalldevs(), because, for example, that process
    does not have sufficient privileges to open them for capturing; if so,
    those devices will not appear on the list.) If pcap_findalldevs() suc-
    ceeds, the pointer pointed to by alldevsp is set to point to the first
    element of the list, or to NULL if no devices were found (this is con-
    sidered success). Each element of the list is of type pcap_if_t, and
    has the following members:
Manual page pcap_findalldevs(3pcap) line 1 (press h for help or q to quit)
```

Chapter 16: Swift on Single Board Computers

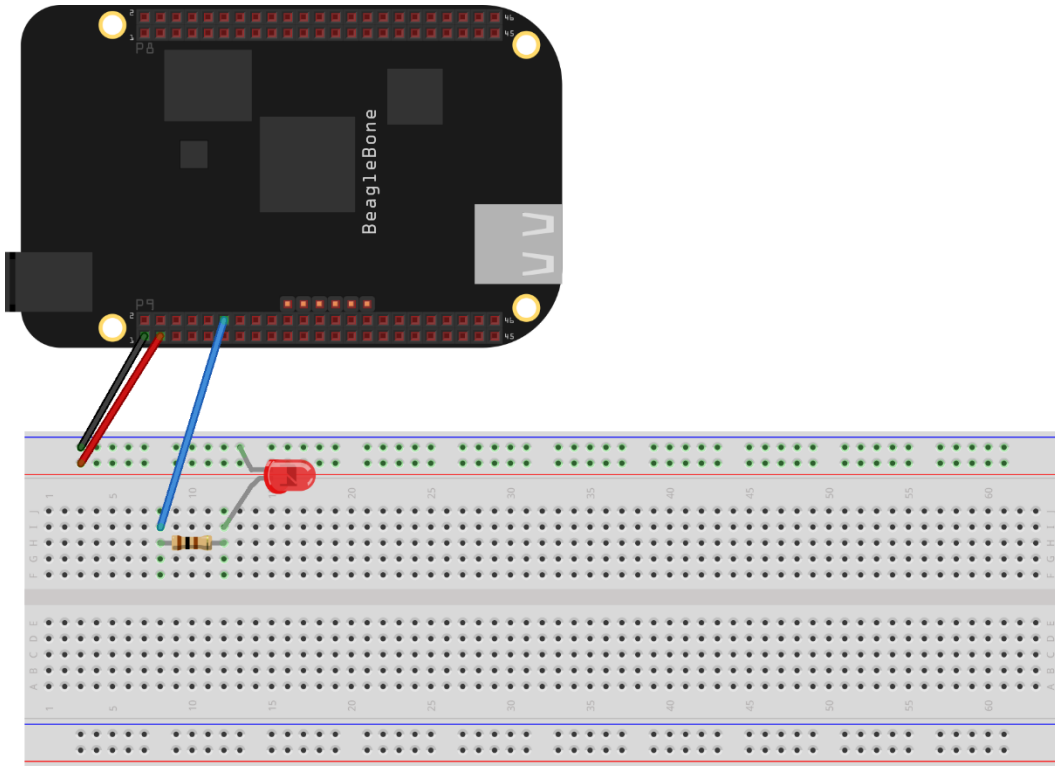




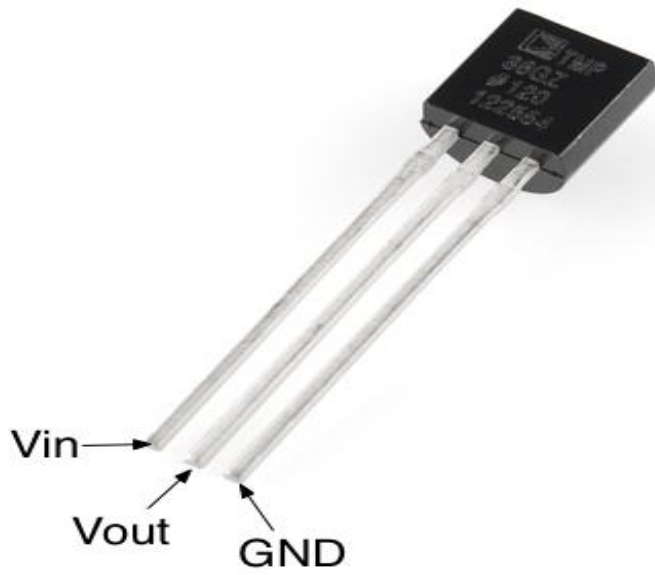
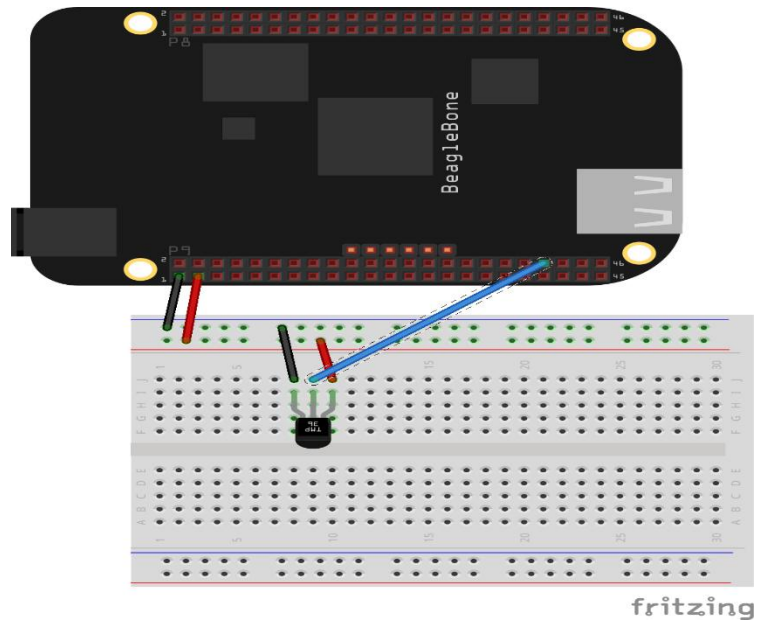


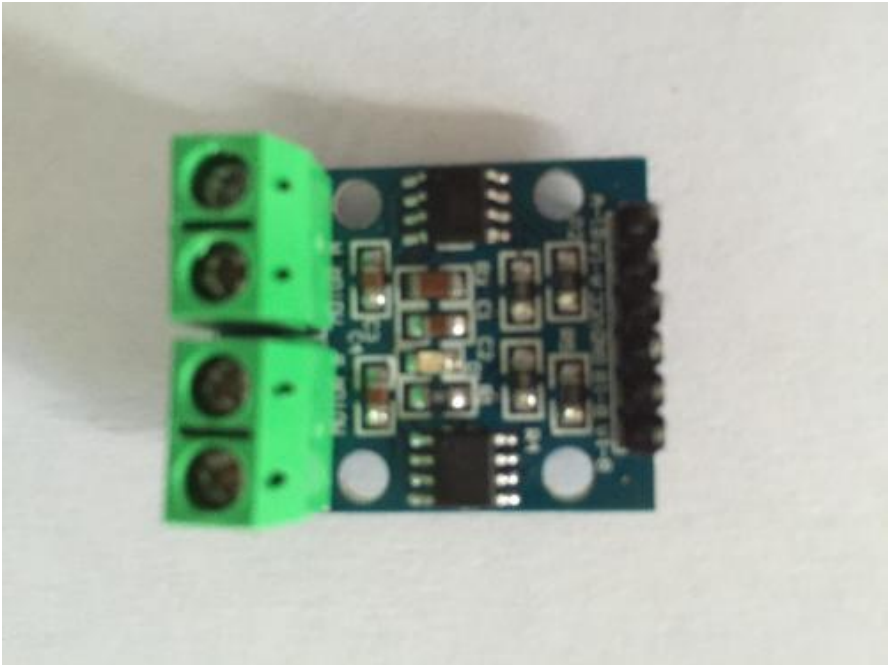
		P8			
		Pin	PIN		
	DGND	1	2	DGND	
	GPIO_38	3	4	GPIO_39	
	GPIO_34	5	6	GPIO_35	
Timer4	GPIO_66	7	8	GPIO_67	Timer7
Timer5	GPIO_69	9	10	GPIO_68	Timer6
	GPIO_45	11	12	GPIO_44	
EHRPWM2B	GPIO_23	13	14	GPIO_26	
	GPIO_47	15	16	GPIO_46	
	GPIO_27	17	18	GPIO_65	
EHRPWM2A	GPIO_22	19	20	GPIO_63	
	GPIO_62	21	22	GPIO_37	
	GPIO_36	23	24	GPIO_33	
	GPIO_32	25	26	GPIO_61	
	GPIO_86	27	28	GPIO_88	
	GPIO_87	29	30	GPIO_89	
	GPIO_10	31	32	GPIO_11	
	GPIO_9	33	34	GPIO_81	
	GPIO_8	35	36	GPIO_80	
	GPIO_78	37	38	GPIO_79	
	GPIO_76	39	40	GPIO_77	
	GPIO_74	41	42	GPIO_75	
	GPIO_72	43	44	GPIO_73	
	GPIO_70	45	46	GPIO_71	

		P9			
		Pin	PIN		
	DGND	1	2	DGND	
	VDD_3V3	3	4	VDD_3V3	
	VDD_5V	5	6	VDD_5V	
	SYS_5V	7	8	SYS_5V	
	PWR_BUT	9	10	SYS_RESET	
	GPIO_30	11	12	GPIO_60	
	GPIO_31	13	14	GPIO_40	EHRPWM1A
	GPIO_48	15	16	GPIO_51	EHRPWM1B
	GPIO_4	17	18	GPIO_5	
	I2C2_SCL	19	20	I2C2_SDA	
EHRPWM0B	GPIO_3	21	22	GPIO_2	EHRPWM0A
	GPIO_49	23	24	GPIO_15	
	FP10_117	25	26	SPI0_14	
	GPIO_125	27	28	GPIO_123	
	GPIO_121	29	30	GPIO_122	
	GPIO_120	31	32	VDD_ADC	
	AIN4	33	34	GND_ADC	
	AIN6	35	36	AIN5	
	AIN2	37	38	AIN3	
	AIN0	39	40	AIN1	
	GPIO_20	41	42	GPIO_7	ECAPPWM0
	DGND	43	44	DGND	
	DGND	45	46	DGND	

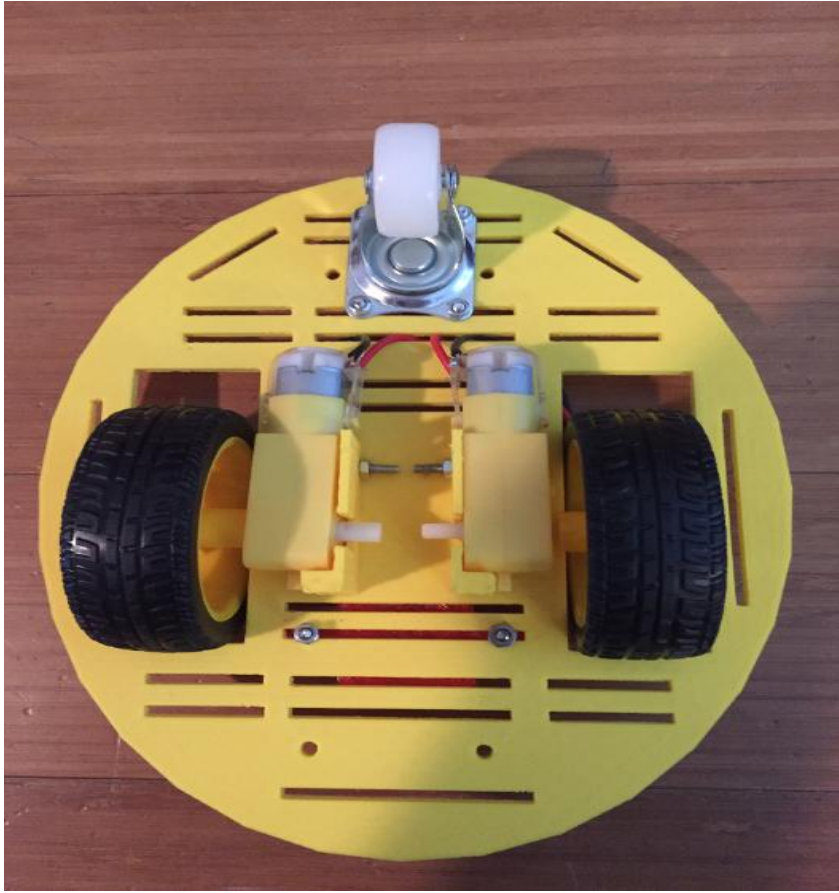


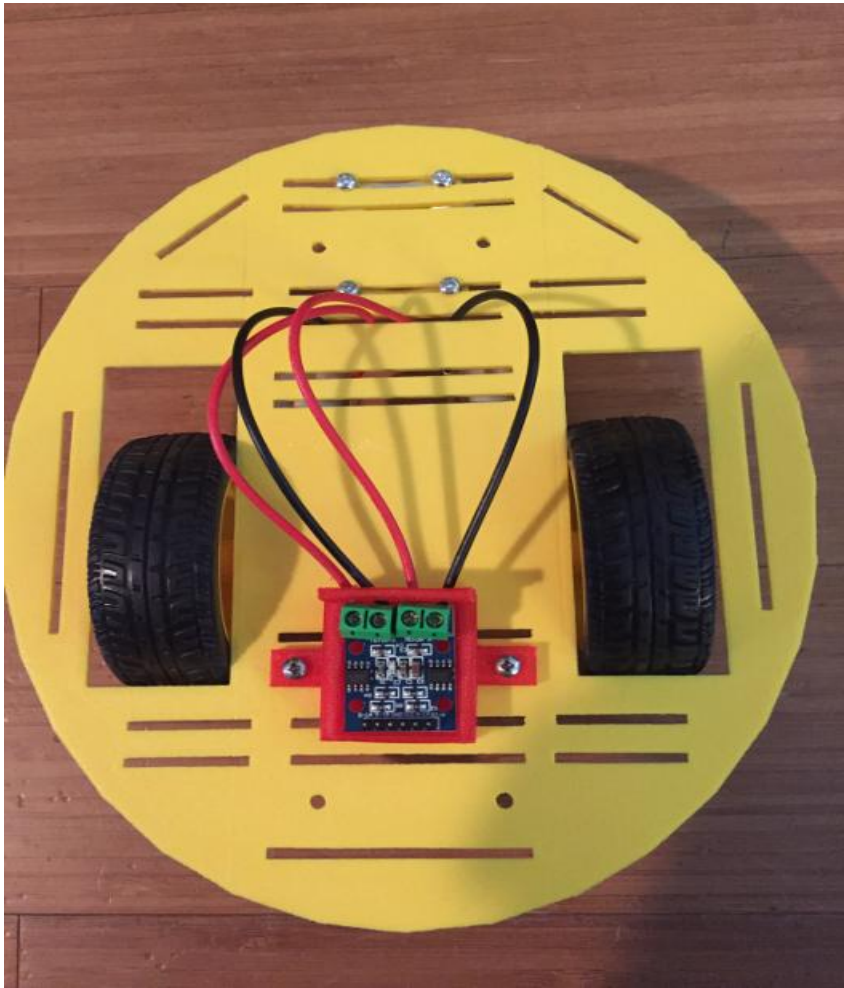
fritzing

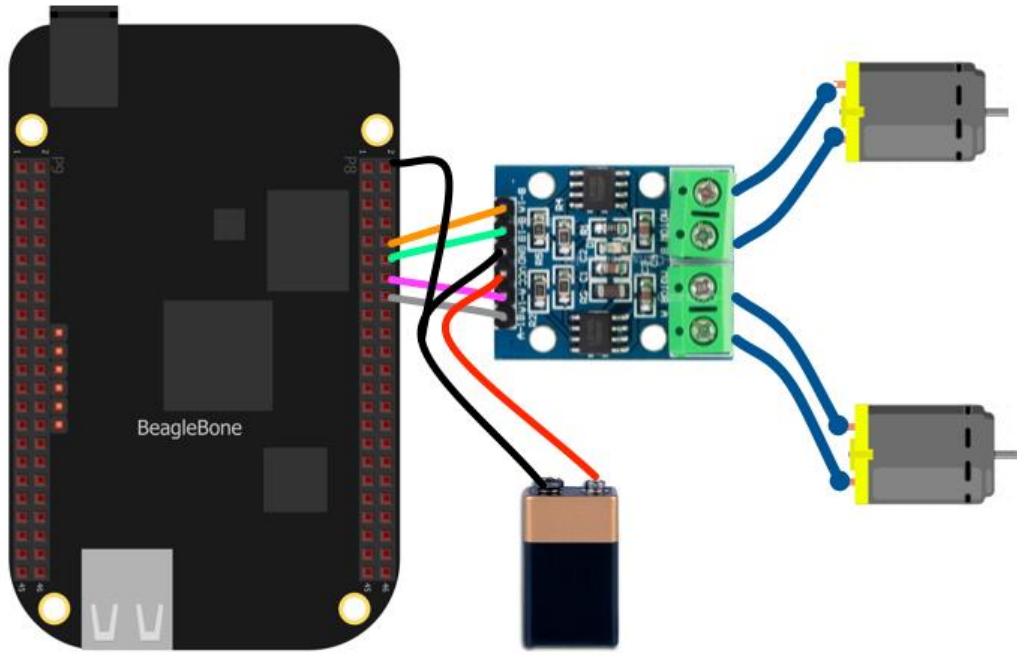


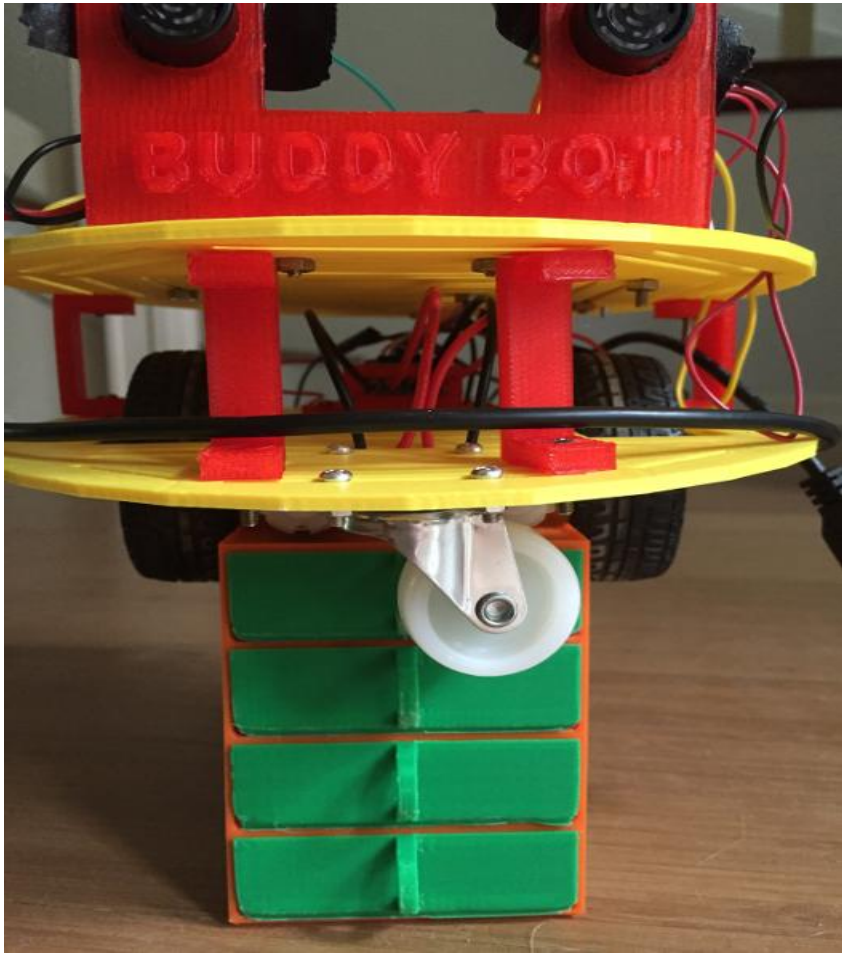


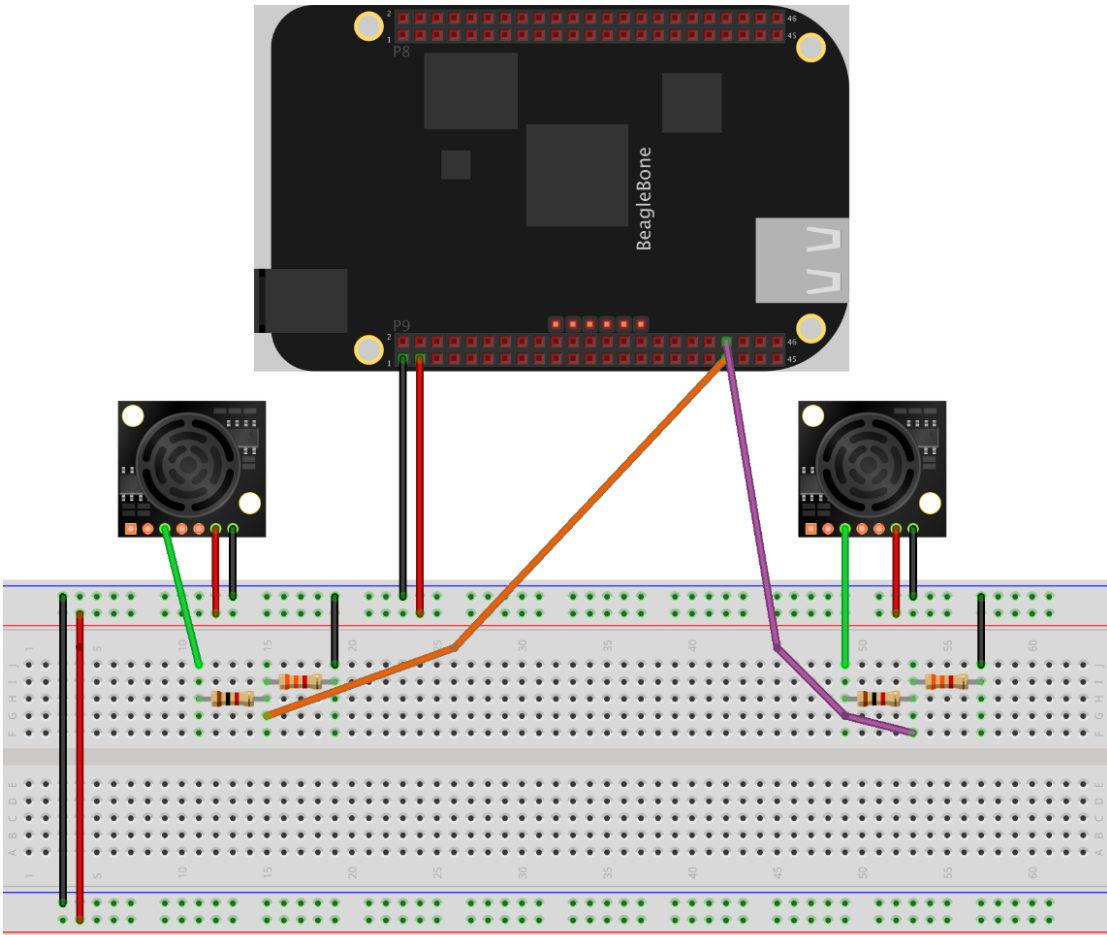












fritzing



