Electronics for IoT

General Purpose Input-Output GPIO

Bernhard E. Boser
University of California, Berkeley
boser@eecs.berkeley.edu

Microcontroller Input/Output

- PC, Laptop
  - Display, Keyboard, Mouse
- Microcontroller
  - I2C … “smart” sensor bus
  - LED …
  - GPIO … direct electrical input / output

Outline

- Digital signals
  - Representation
  - Computing
- Digital I/O
  - Output
  - Input
  - Interrupts
- Analog I/O

Binary Signals

Switches

Transistors

Example: Inverter

Chain of Inverters

Noise

Noise Margin

Binary Signals

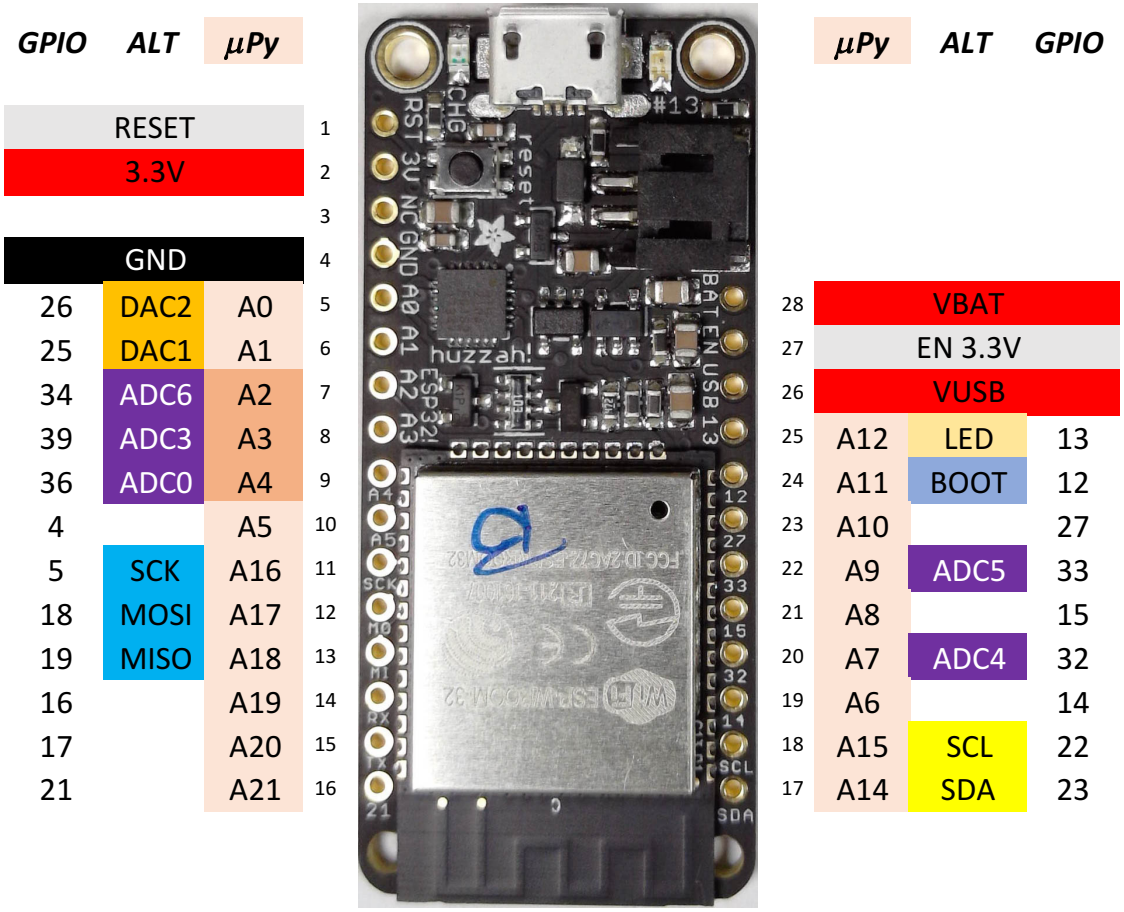
Binary Computation: Example - NOR

What about OR?

What about … addition?

Digital GPIO

- Directly represent internal states at chip pins

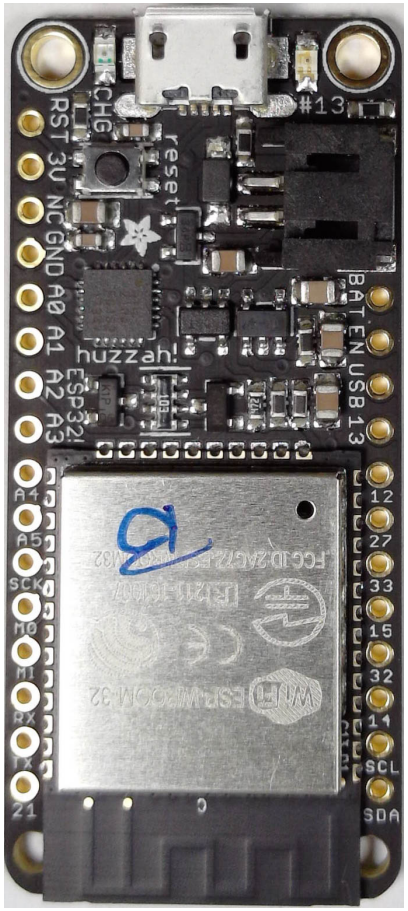


board.py

```
from micropython import const

# HUZZAH32 pin definitions
A0 = const(26)
A1 = const(25)
A2 = const(34)
A3 = const(39)
A4 = const(36)
A5 = const(4)
A6 = const(14)
A7 = const(32)
A8 = const(15)
A9 = const(33)
A10 = const(27)
A11 = const(12)
A12 = const(13)
A14 = const(23)
A15 = const(22)
A16 = const(5)
A17 = const(18)
```

GPIO	ALT	µPy		
RESET			1	
3.3V			2	
			3	
GND			4	
26	DAC2	A0	5	
25	DAC1	A1	6	
34	ADC6	A2	7	
39	ADC3	A3	8	
36	ADC0	A4	9	
4		A5	10	
5	SCK	A16	11	
18	MOSI	A17	12	
19	MISO	A18	13	
16		A19	14	
17		A20	15	
21		A21	16	



µPy	ALT	GPIO	
28	VBAT		
27	EN 3.3V		
26	VUSB		
25	A12	LED	13
24	A11	BOOT	12
23	A10		27
22	A9	ADC5	33
21	A8		15
20	A7	ADC4	32
19	A6		14
18	A15	SCL	22
17	A14	SDA	23

GPIO Pins

- Most pins
  - Digital input or output
- Some special functions
- LED internally connected

Adafruit HUZZAH32 MicroPython

GPIO	ALT	µPy		µPy	ALT	GPIO
RESET			1			
3.3V			2			
			3			
GND			4			
26	DAC2	A0	5			
25	DAC1	A1	6			
34	ADC6	A2	7			
39	ADC3	A3	8			
36	ADC0	A4	9			
4		A5	10			
5	SCK	A16	11			
18	MOSI	A17	12			
19	MISO	A18	13			
16		A19	14			
17		A20	15			
21		A21	16			
				28	VBAT	
				27	EN 3.3V	
				26	VUSB	
				25	A12	LED 13
				24	A11	BOOT 12
				23	A10	27
				22	A9	ADC5 33
				21	A8	15
				20	A7	ADC4 32
				19	A6	14
				18	A15	SCL 22
				17	A14	SDA 23

Boot mode:

BOOT (A11) has a built-in pull-down
Connect to 3.3V on power-up for safe boot.

EN 3.3V: tie to GND to disable regulator

Legend:

sup	ADC	SPI
GND	DAC	I2C
BOOT	VBAT/2	LED
input only!		
VBAT/2 tied to VBAT2		

Digital Output

GPIO with MicroPython

https://github.com/bboser/IoT49

IoT49 Micropython Programming Setup

ESP32 MicroPython programming setup for UC Berkeley course EE49, Electronics for IoT.

Installation

- Instructions
- Atom IDE (optional)

Documentation

- GPIO (General Purpose Input/Output)
  - HUZZAH32 pin diagram
  - Digital
  - Analog
- IOT49 Firmware for HUZZAH32
  - Micropython
  - Built-In Modules
  - MicroPython standard library
  - Source Code

Digital Output

Standard Output

In standard mode, the pin is driven to GND (0V) or VDD (3.3V) depending on its state.

```
from machine import Pin
p = Pin(id, mode=Pin.OUT)
p(0)   # pin driven to 0V
p(1)   # pin driven to VDD (~ 3.3V)
```

`id` is the name of the pin, e.g.

```
from board import A0
from machine import Pin
p = Pin(A0, mode=Pin.OUT)
```

Open Drain Output

Open Drain Example: I2C

Example: LED

Summary

- Microcomputers use binary signals internally
- Transistors act like switches
- Digital signals are inherently error tolerant
- Digital GPIO
  - Electrical bit-wise input/output
  - Standard, open-drain output