Internet of Things: Some exercises

John L. Manferdelli

johnmanferdelli@hotmail.com

April 6, 2020, 13:40

Example exercises

1. Use an Arduino to measure light intensity (roughly) using a photo resistor.
2. Use an Arduino to measure temperature (roughly) using a thermistor.
3. Modify configuration information and update for, say a camera.
4. Modify firmware on an Arduino. How about an ARM based processor.
5. Map the boot sequence on an Arduino and an ARM based processor.
6. Use an Arduino to get position on earth using GPS.
7. Use a pair of Arduinos to transmit/receive wireless information using private channel (HC-12).
8. Reverse engineer a piece of software (see Troy's exercise).
9. Log into an IP camera, change parameters (see the reverse engineering exercise).
10. Modify a router's OS/firmware.
11. Use updated SW to implement a side channel using an LED, speaker.
12. Design an Arduino based system to collect information from devices.
13. Develop an optical communication link using an LED or laser and an optical transistor.

Example exercises

14. Extract encrypted passwords from /etc/shadow and use a password cracker to find them.
15. Interrupt boot on u-boot to become root and change files.
16. Develop an entropy measurement and use it to find keys in an image.
17. Discover the "roots of trust" (public/private keys) embedded in an image and modify them.
18. Develop a mechanism to spoof a GPS signal. DO NOT TEST THIS without talking to a lawyer to ensure legal compliance. You will, at a minimum need to do tests in a Faraday cage. Seriously, don't do it.
19. MITM an IoT device via a router (a camera should work). What does it talk to? What does it transmit?
20. Map the update mechanism of an IoT device.
21. How to update firmware and OS when embedded root CA is expired.
22. Map the IoT devices in [your house, the IoT lab, a hotel, a manufacturing floor]
23. Desolder a ROM and read the image.
24. Find a "fingerprint" for an IoT device? Can you measure it remotely?

Exercises 14, 15, 16,17: see ”Breaking an IP camera”. For 19, see Reverse engineering section. For 23, see corresponding exercise in Reverse engineering section.

Example exercises

23. What are the available sources of entropy in an IoT device? At what rate could you generate AES-256 bit keys with such entropy?
24. Map all the important configuration files on an IoT system.
25. Do some of the exercises we did with the Arduino with a Raspberry Pi.
30. Stimulate a glitch on a pin using a few SDRs in close proximity to a known board/chip.
31. Add capacitance to a interface that makes measurements inaccurate. How would you detect HW trojans in an IoT device?
32. Estimate the RF emanations from an IoT device? How would you shield them?
33. Find a moderate cost tamper evidence system and estimate the cost to defeat it. How would you use scale to help limit the risk?
34. Intercept and read an 802.11 message using an SDR.
35. Jam an 802.11 message.
36. Locate an 802.11 emitter using a directional antenna (coffee can or pringles can depending on band).

Example exercises

37. Measure g (the gravitational constant near Earth's surface) with optical transistors and small lasers (about $5).
38. GPS/dead reckoning, now do it with RTL-SDR and HackRF One (you'll need an am and filter).
39. Design and build a weather station.
40. Make the weather station communicate with a computer in the house (say, using NFC).
41. Jam the protocol using an HackRF One or other SDR.
42. Build a Morse code detector and decoder (using light and sound).
43. Detect when your dog is barking.
44. Simulate a buffer overflow attack.
45. Reverse engineer a child's toy (say a remote controlled tank).
46. Scare your kid by taking control of it.
47. Figure out where you are with a GPS sensor and verify it with other means.
48. Design and implement a GPS based route planner with waypoints.

Example exercises

49. Measure the takeoff speed of plane with an accelerometer.
50. Detect people in classroom. Can you count them?
51. Determine when to water the plants.
52. Design a tool guider with a ping sensor.
53. Record sunrise/sunset with a sensors.
54. Pick some code, fuzz, find a buffer overflow, do a code redirect.
55. Use TFTP to download an exploit kit (TO YOUR MACHINE ONLY).
56. Reverse engineer a script (like the reverse shell).
57. Find a key in firmware, change it and repackage the FW.
59. Do the exercises in the addendum to the reverse engineering section
60. Develop a side channel (RF with SDR)
61. Detect when a is plane landing.
62. Detect when a car stops.

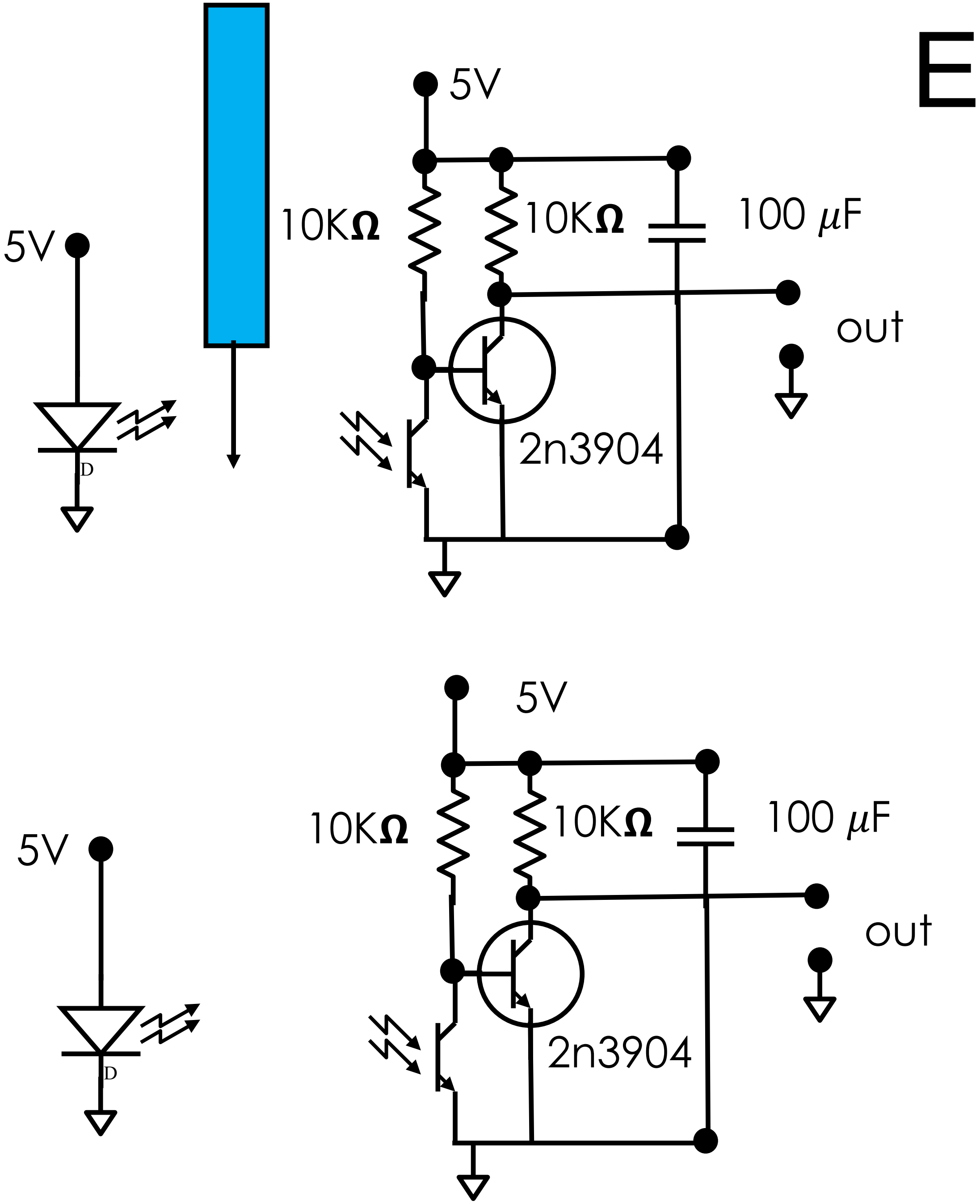
Example exercises

63. Determine if you accelerating/decelerating too quickly.
64. Decode a telephone number from the sounds.
65. Build a coded entry system.
66. Design and test an IR communication channel.
68. Read a Flash. Reverse engineer the file system code.
69. Hit the beach? Determine what the sun exposure is today.
70. Count cars. (time of entry exit? distinguish between entry/exit).
71. Have an existing camera send pictures elsewhere.
72. Track garage doors in you neighborhood.
73. Determine patterns of life based on temperature, sound, light (sleep, watch TV, home, gone).
74. Track a route via altitude (works in SF, not Iowa).
75. Detect RF emissions from a board.

Example exercises

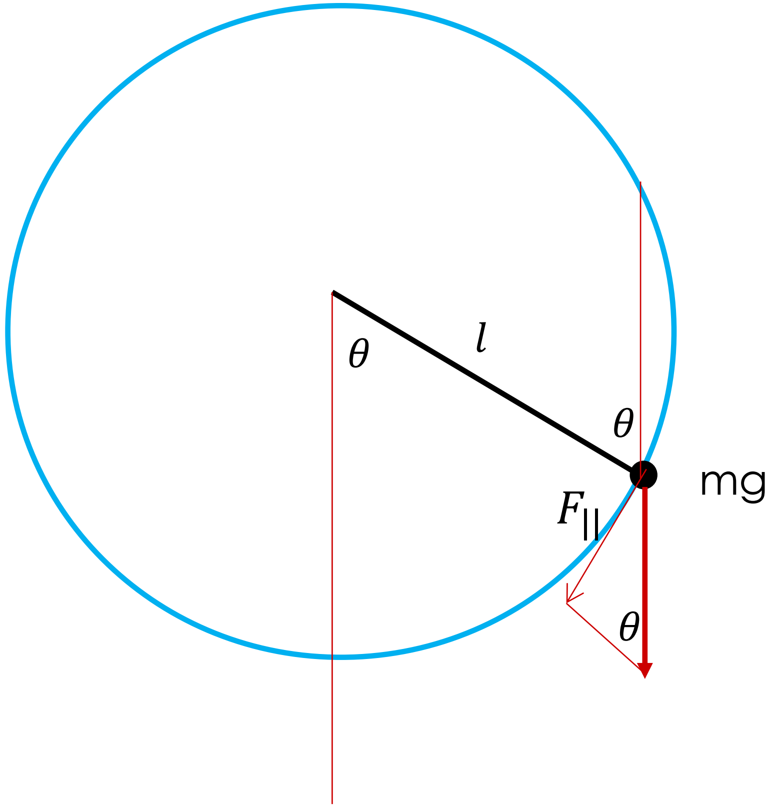
76. Build a blimp with radio control and auto pilot to waypoints. (See Ressler, Do it yourself engineering, The Great Courses.)
77. Listen to an FM station with and SDR.
78. Listen and display weather from a NOAA broadcast with SDR.
79. Slurp wireless traffic with an SDR.
80. Listen to ADSB (air traffic control) with RTL-SDR.
81. We use a lot of libraries that hide the details of the GPIO interfaces like WiringPi, SoftwareSerial, and others. These are all open source. Pick one and figure out the interface details.
82. Design a "low jack" system for your car, plane or bicycle. It should detect unauthorized movement and report position.

Exercise



- Measure g
  - Opaque slab drops.
  - Each output goes to an Arduino analog pin. The output is sampled every 10μs.
  - For light source, I used a laser focused on optical transistor.
  - Outputs go high when slab blocks light source.
  - We use the output timings to figure out how long the slab took to fall.
  - It’s best to do the experiment in a dark room.

Exercise



- $F_{||} = mgsin(\theta) \approx mg\theta$
- $ml\frac{d^2\theta}{dt^2} = -mg\theta$
- $\theta = Asin(\omega t), \omega = \sqrt{\frac{g}{l}}$
- $T = 2\pi\sqrt{\frac{l}{g}}$
- Place laser/detector on previous slide at bottom of arc ($\theta = 0$)
- Time successive swings of mass though ($\theta = 0$), this is just T.
- Knowing T and l, use formula above to calculate g.

l (m)	T (sec)
.1	.6344
.2	.8971
.3	1.0988

Exercise --- HC-12

```
// hc12
// Manferdelli

#include <SoftwareSerial.h>

const int deviceReceivePin= 4;
const int deviceTransmitPin= 5;
const int deviceSetPin = 6;
#ifndef byte
typedef uint8_t byte;
#endif

// Note: device transmit pin is SoftwareSerial receive pin
//   and vice-versa.
SoftwareSerial hc12(deviceTransmitPin, deviceReceivePin);

void copy(char* from, char* to, int size) {
  for(int i = 0; i < size; i++)
    to[i] = from[i];
  return;
}
```

Exercise

```
int read_from_serial(int max, char* b) {
  int i = 0;
  while (hc12.available() != 0 && (i < max)) {
    b[i++] = hc12.read();
  }
  return i;
}

void setup() {
  pinMode(deviceSetPin, OUTPUT);
  delay(100);
  Serial.begin(9600);
  hc12.begin(9600);
}
```

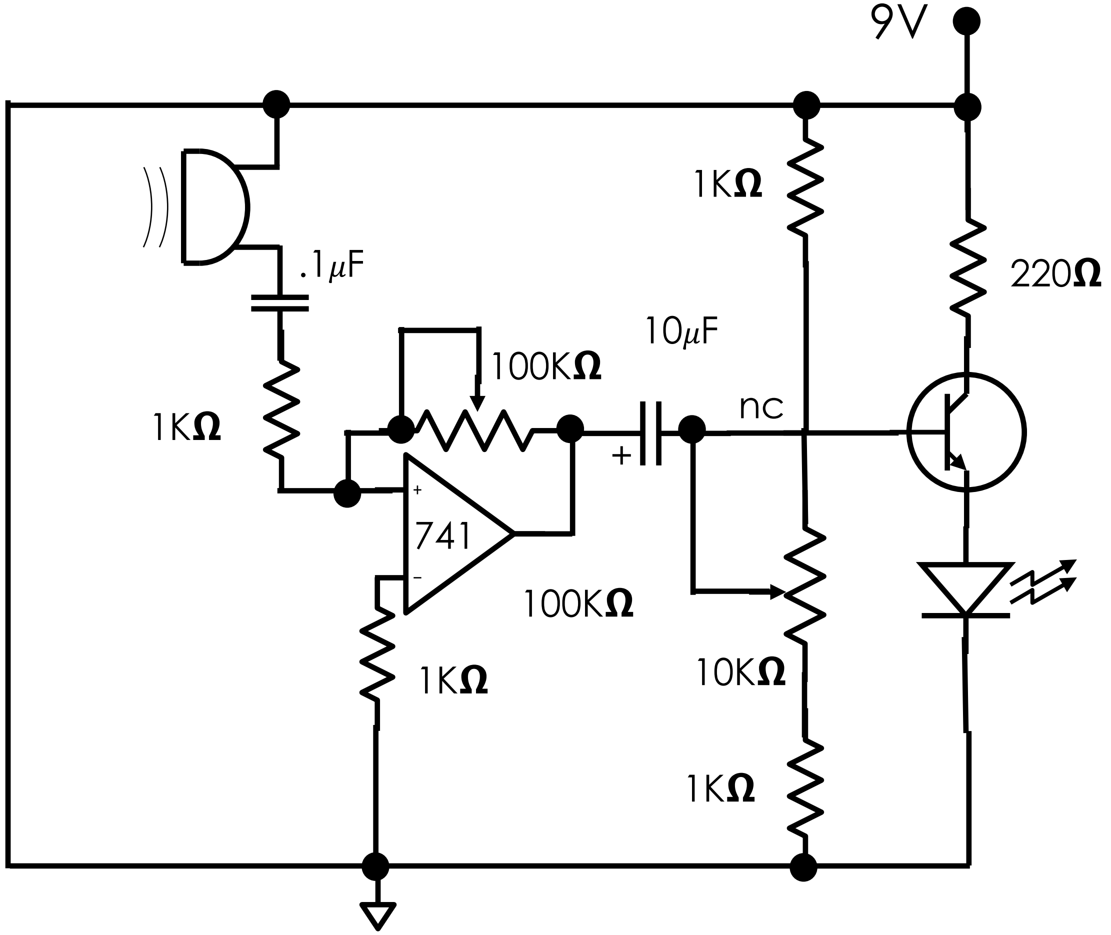
```
void loop() {
  char send_buf[65];
  char receive_buf[65];
  copy((char*)"AT", send_buf, 3);
  for (;;) {
    digitalWrite(deviceSetPin, HIGH);  // Transparent mode
    delay(200);
    hc12.listen();
    int n = read_from_serial(64, receive_buf);
    if (n > 0) {
      receive_buf[n] = 0;
      Serial.print("Received: ");
      Serial.println((const char*)receive_buf);
    } else {
      Serial.println("nothing received");
    }
    digitalWrite(deviceSetPin, LOW); // Command mode
    delay(500);
    hc12.print(send_buf);
    Serial.print("Sent ");
    Serial.println((const char *)send_buf);
  }
}
```

Exercise

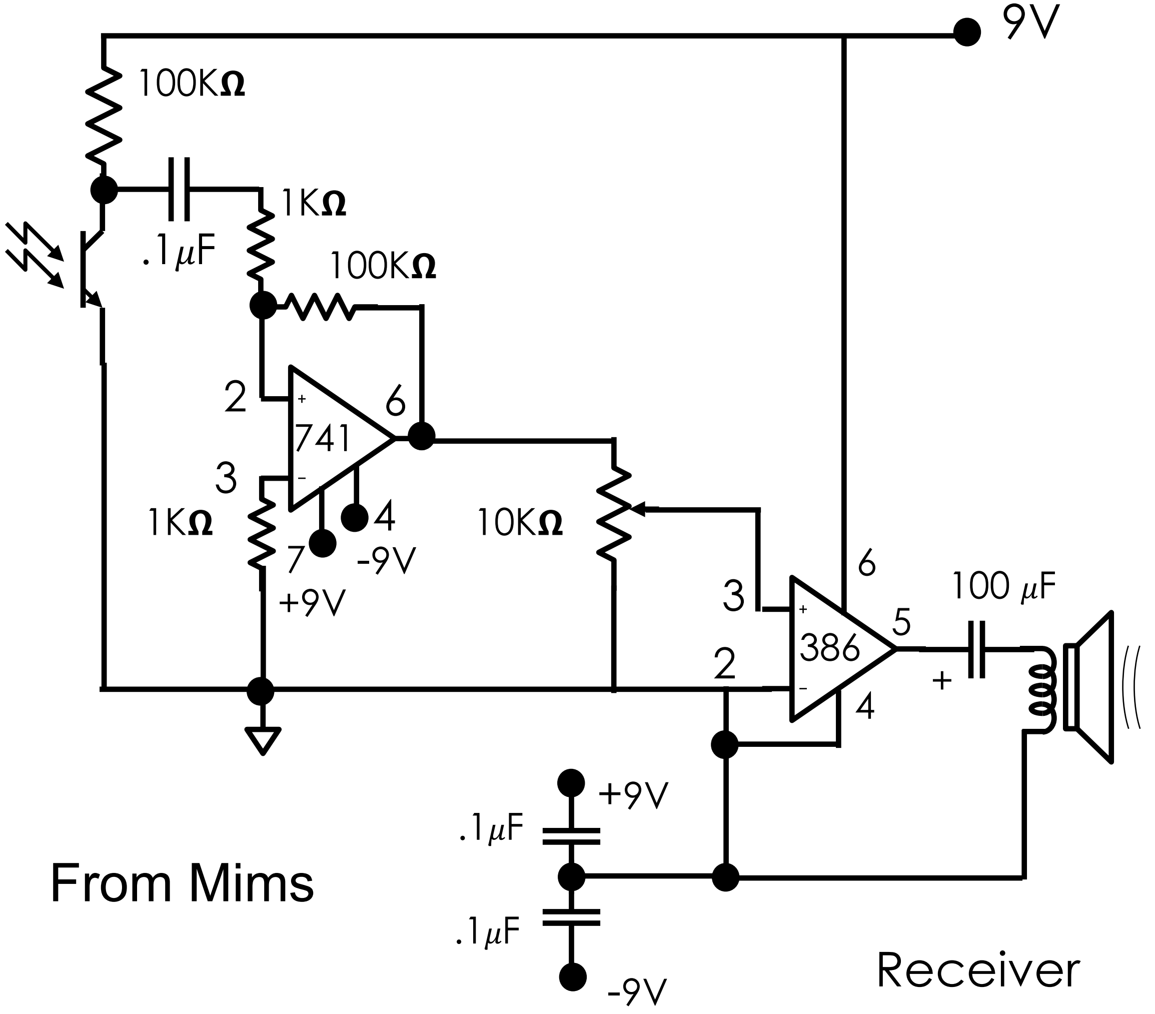
- Optical Communication
  - Analog: voice
  - Digital: PWM
  - Digital: UART

Transmitter

From Mims



Exercise



From Mims

- Optical Communication
  - Analog: Voice
  - Digital: PWM
  - Digital: UART

Exercise

- Raspberry Pi exercises
  - You’ll need Wiring Pi which is now a standard library on Raspberry Pi.
  - See this site.

Exercise

- Stimulate a glitch on a pin using a few SDRs in close proximity to a known board/chip.

Exercise

- Add capacitance to a interface that makes measurements inaccurate.
- How would you detect HW trojans in an IoT device?

Exercise

- Estimate the RF emanations from an IoT device?
- How would you shield them?

Exercise

- Find a moderate cost tamper evidence system and estimate the cost to defeat it.
- How would you use scale to help limit the risk?

Exercise

- Intercept and read an 802.11 message using an SDR.
    - You’ll need an SDR and an OFDM demodulator. See this paper.

Exercise

- Jam an 802.11 message.

Exercise

- Locate an 802.11 emitter using a directional antenna.
    - Use coffee can or pringles can depending on band for directional antenna.

Exercise

- GPS/dead reckoning, now do it with RTL-SDR and HackRF One (you’ll need an amp and filter).

Exercise

- Design and build a weather station.
  - We did individual sensors in the electronics section.
  - Maybe use I^2C?
- Make the weather station communicate with a computer in the house (say, using NFC).
- Jam the protocol using an HackRF One or other SDR.

Exercise

- Build a Morse code detector and decoder (using light and sound).
  - You can find a tutorial here.

Exercise

- Reverse engineer a child’s toy (say a remote controlled tank).
- Scare your kid by taking control of it.

Exercise

- Design and implement a GPS based route planner with waypoints, say, for a drone.
- Distance (optimistic): 20 minutes-power x 22 meters/sec x 60 sec/min is about 26 km

- Here’s (approximately) a route I’d be interested in.

  - Home: 37.759, -122.439, elevation: 300 ft
  - Exploratorium: 37.800, -122.358
  - Maybe visit the rock?
  - North end of treasure island: 37.831, -122.375
  - Berkeley pier: 37.864, -122.313
  - Evans Hall, UCB: 37.874, -122.258, elevation: 400 ft

  - Watch out for tall buildings and hills
  - Sutter office building is right on route
  - Height of Salesforce tower: 1074 ft.
  - Mt Sutro: 900 ft

Exercise

- Record sunrise/sunset with a sensors.

Exercise

- Pick some code, fuzz, find a buffer overflow, do a code redirect.
  - We showed how to fuzz in the software slides.
  - Maybe use Ghidra to reverse engineer the code?

Exercise

- Use TFTP to download an exploit kit

Exercise

- Reverse engineer a script (like the reverse shell).

Exercise

- Find a key in firmware, change it and repackage the FW.

Exercise

- Develop a side channel (RF with SDR)

Exercise

- Detect when a is plane landing.
- Detect when a car stops.
    - See the accelerometer exercises in the Electronics section.

Exercise

- Determine if you accelerating/decelerating too quickly.
    - See the accelerometer exercises in the Electronics section.

Exercise

- Decode a telephone number from the sounds.
  - There is an tutorial here.
  - Now how about arbitrary tones?

Exercise

- Build a coded entry system.
  - You should consult the “keypad” exercise in the Electronics section.

Exercise

- Design and test an IR communication channel.
    - See the IR sensor exercise in the Electronics section.

Exercise

- Read a Flash. Reverse engineer the file system code.
    - See the exercise and documentation in the Reverse Engineering section.

Exercise

- Count cars. (time of entry exit? distinguish between entry/exit).

Exercise

- Have an existing camera send pictures elsewhere.
  - Consult the “breaking an IP camera” exercise in the Reverse engineering section.

Exercise

- Track garage doors in you neighborhood.
    - The Hack RF One tutorials mentioned earlier has a section on this.

Exercise

- Determine patterns of life based on temperature, sound, light (sleep, watch TV, home, gone).
  - Scary, huh?

Exercise

- Track a route via altitude (works in SF, not Iowa).
  - See the barometric pressure exercises in the Electronics section.
  - It’s much harder finding elevation than location information.

Exercise

- Build a blimp with radio control and auto pilot to waypoints. (See Ressler, Do it yourself engineering, The Great Courses.)
    - You can use the toy controller for the tank to control the blimp so you can use the same SDR stuff.

Exercise

- Listen and display weather from a NOAA broadcast with SDR.

Exercise

- Slurp wireless traffic with an SDR.
    - See earlier 802.11 exercises

Exercise

- Listen to an FM station with an SDR.
  - This is the first exercise in the HackRF Tutorials here.

Exercise

- Listen to ADSB (air traffic control) with RTL-SDR.
  - There is a tutorial here.

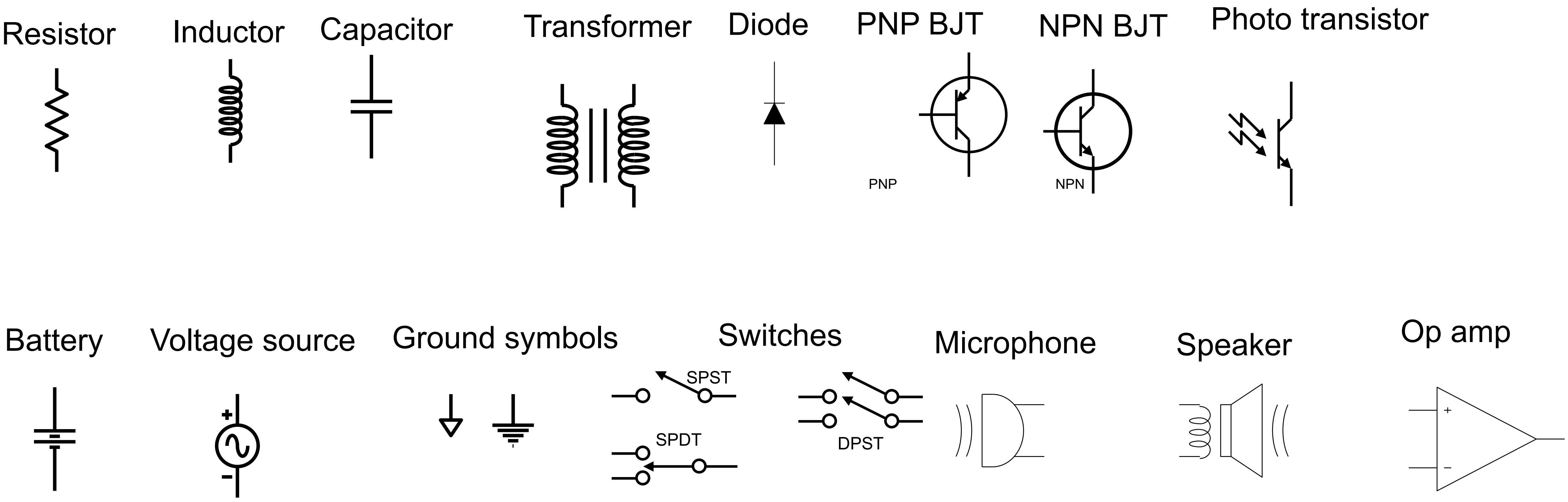
Exercise

- We use a lot of libraries that hide the details of the GPIO interfaces like WiringPi, SoftwareSerial, and others. These are all open source. Pick one and figure out the interface details.

Exercise

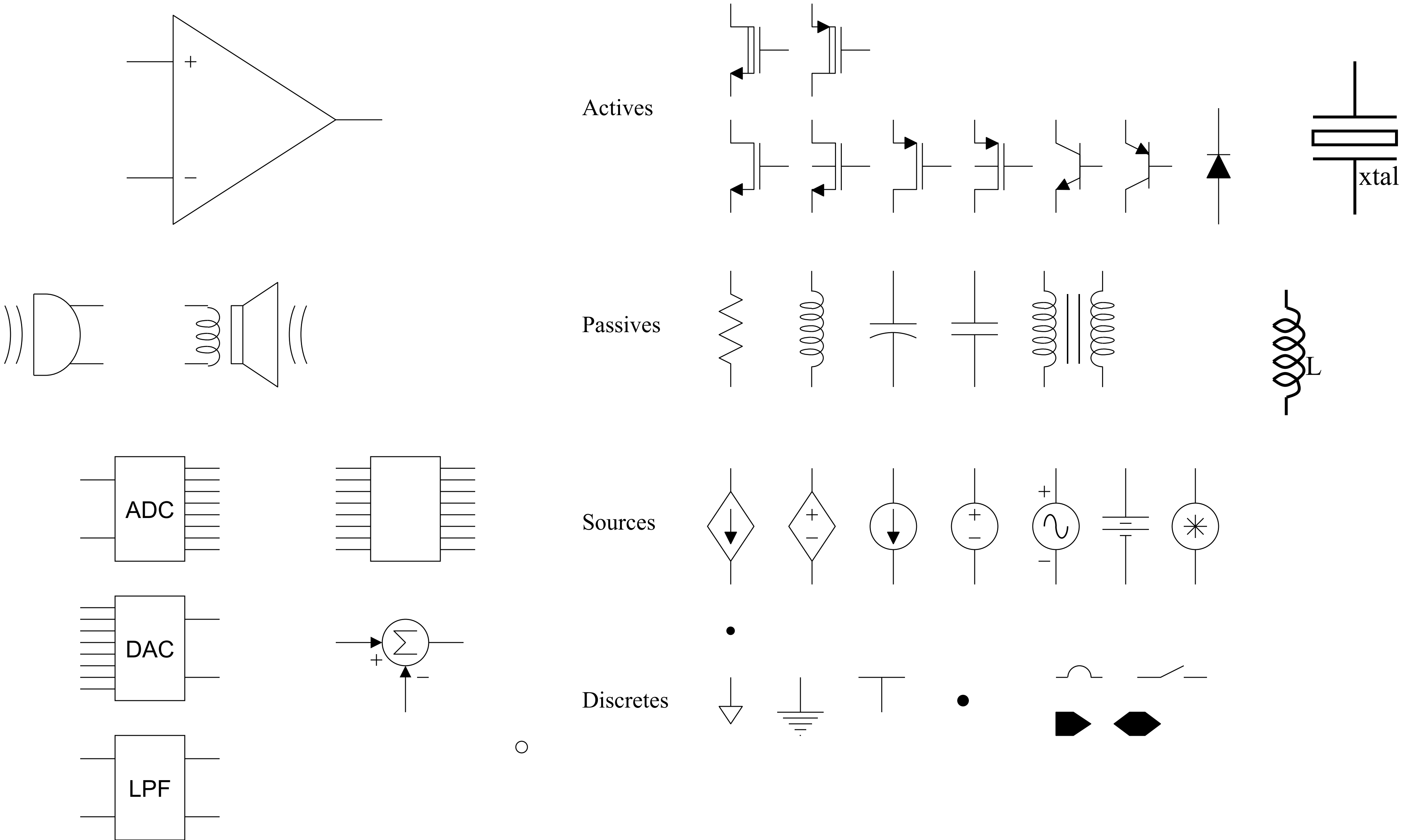
- Make up some exercises of your own and tell me so I can include them.

Circuit symbols



Symbols from http://www.eehomepage.com/symbols.ppt. Copyright 2008 by EE HomePage.com. These symbols may be freely reused.

Circuit symbols



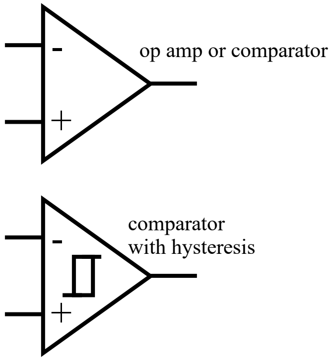
Actives

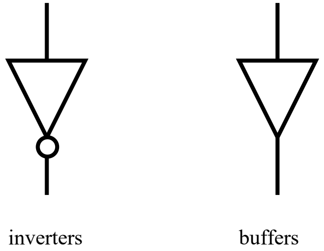
Passives

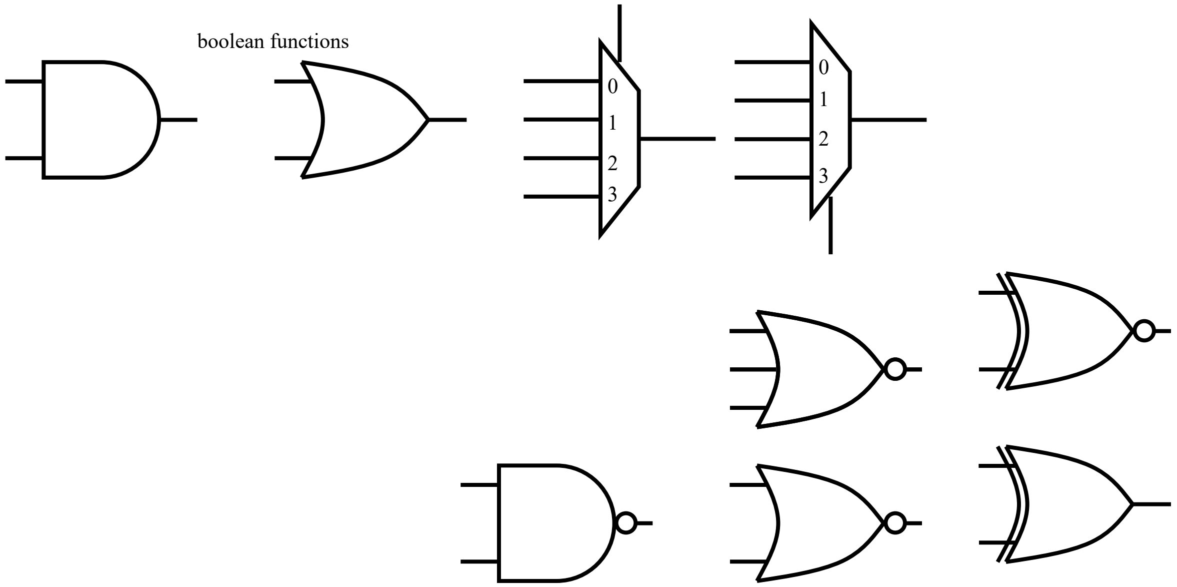
Sources

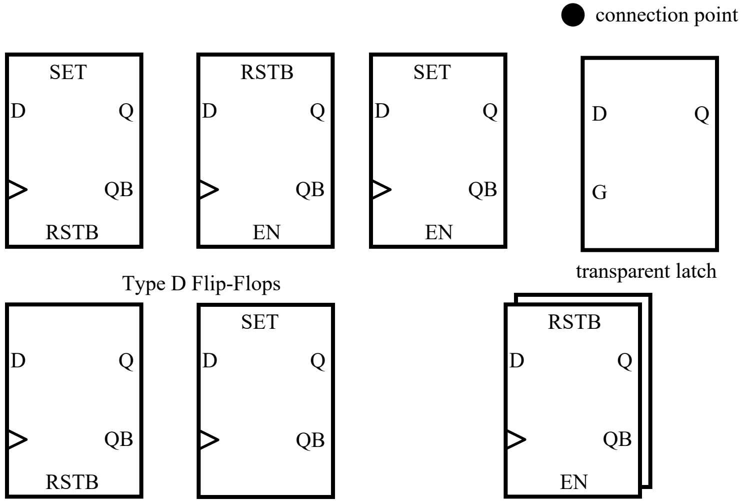
Discretes

Circuit symbols



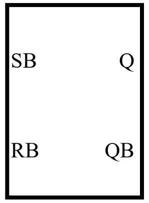




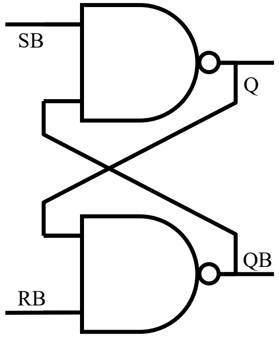


http://www.eehomepage.com/symbols.ppt

Circuit symbols



$\overline{\text{SR}}$ NAND Latch



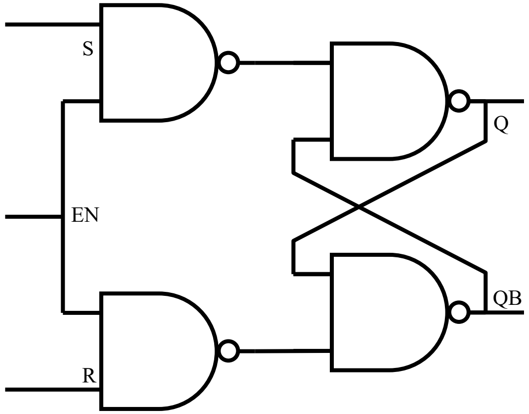
SB	RB	Action

0	0	Q=1
		QB=1

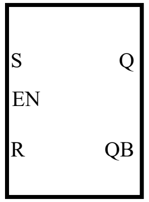
0	1	Q=1
		QB=0

1	0	Q=0
		QB=1

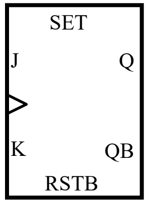
1	1	Keep state



EN	Action
0	Keep state
1	same as SR latch



Gated SR Latch



JK Flip Flop

J	K	Next Value
0	0	No Change
0	1	Q=0
1	0	Q=1
1	1	Toggle