

CONCEPT

Arduino for Ham Radio



Objectives

- Start from where you are... learn step-by-step
- Using your own Laptop
 - Revise from last course
 - Or start at the beginning
- True hands-on amateur radio, you contribute
- Build VFO and SDR receiver shields



Concept

- S1 - Arduino, Learner
- S2 - VFO & RTC
- S3 - Eagle PCB design
- S4 - ROTARY ENCODER & LCD
- S5 - Build the VFO kit
- S6 - SDR design
- S7 - Build the SDR kit





New to Arduino?

Let's start at the beginning.
Apologies to experienced users!

Experiment board

To start you need:

- A Windows PC or a Mac
- Arduino, Breadboard, wires, etc (Kit 1)
- LED & 220R
Piezo active buzzer (Incl in Kit 2)



Kit 1 & 2

	Starter	Learner	VFO & RTC	VFO+ROT	VFO+ROT+LCD
Arduino UNO	x				
400 point BB	x				
Jumper wires	x				
LED		x			
220R		x			
Piezo buzzer		x			
Si5351 module			x	o	o
Rotary Encoder				x	o
I2C LCD			x		o
RTC module			x		
CR1220 battery			x		
F - M wires			x		o

Check your kits now



Kit 1

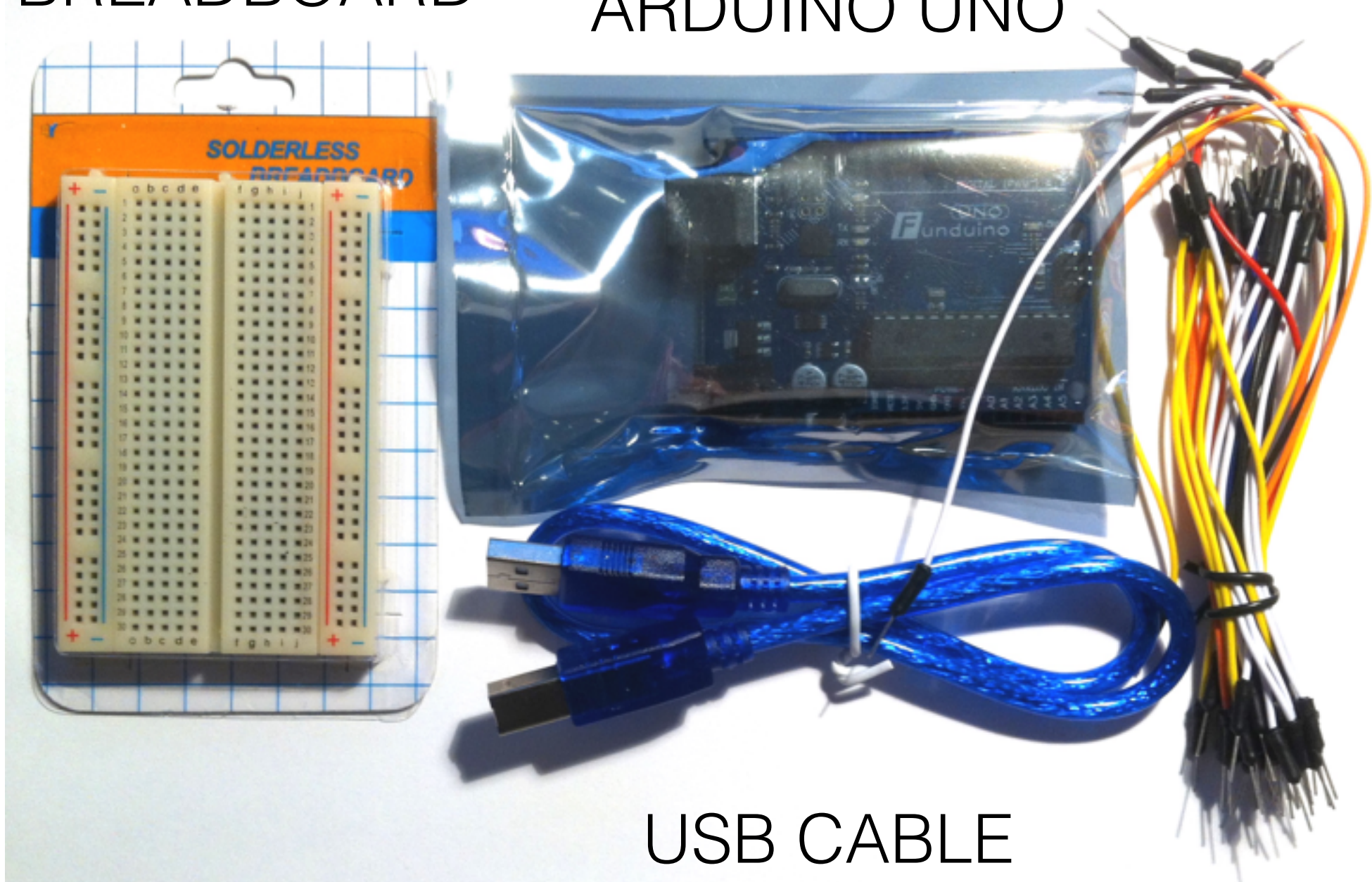
Plus FREE
mounting plate

BREADBOARD

ARDUINO UNO

WIRES

USB CABLE

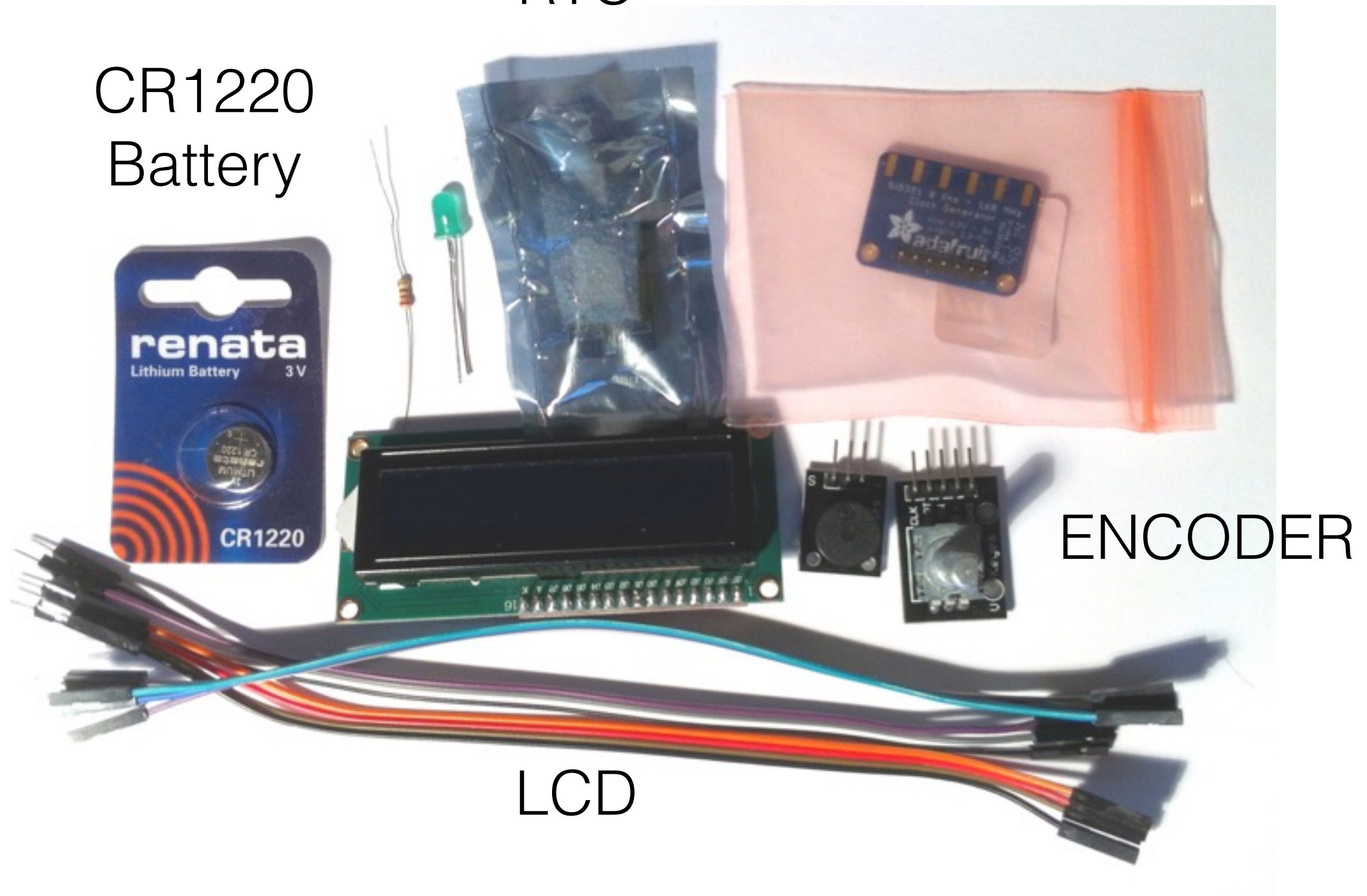


Kit 2

RTC

VFO

CR1220
Battery



ENCODER

LCD

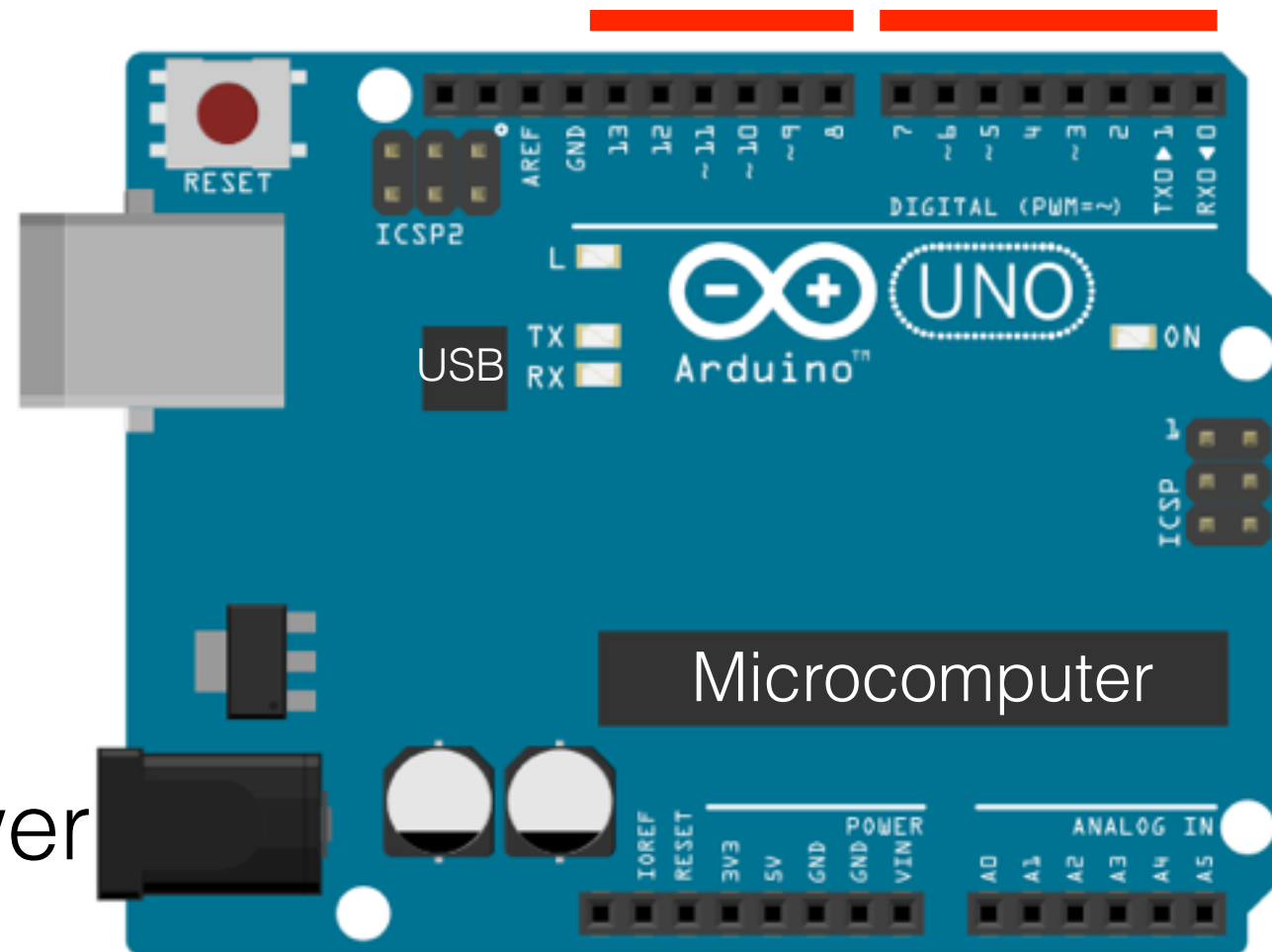


The Arduino UNO

Digital Input/Outputs D13-D0

USB
From PC
+5V power

+7-12V power



Power Outputs & Analog Inputs A0-A5 (or Digital I/O)

A4 & A5 = I2C serial bus SDA & SCL



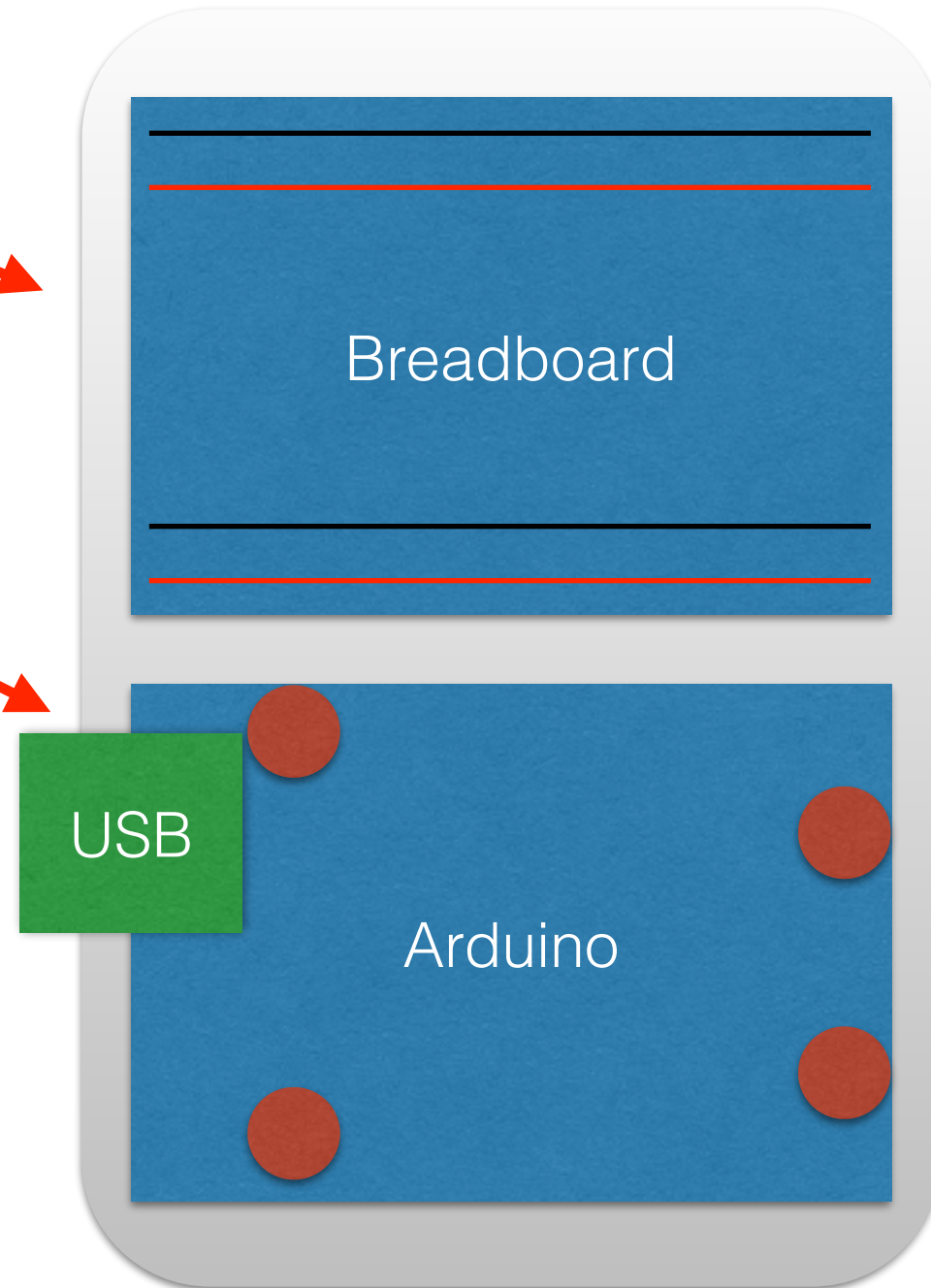
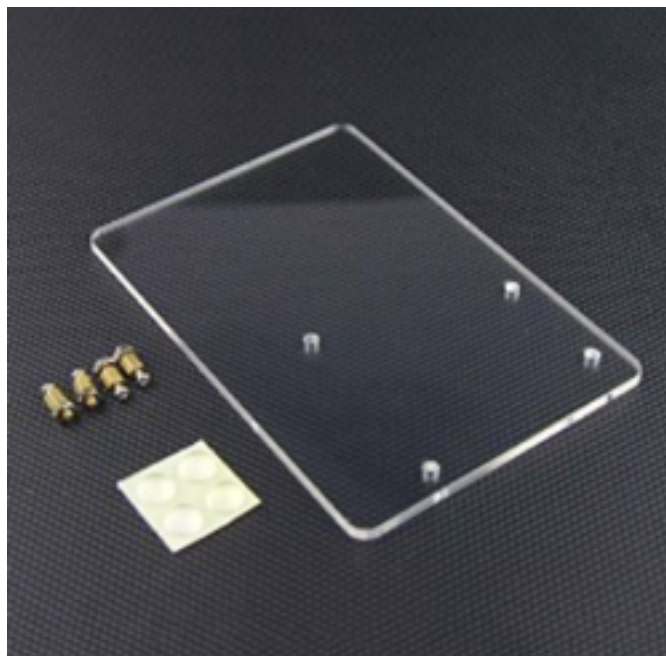
Build your starter kit

- Stick on the breadboard

2

- Fix the Arduino

1



Black
Red



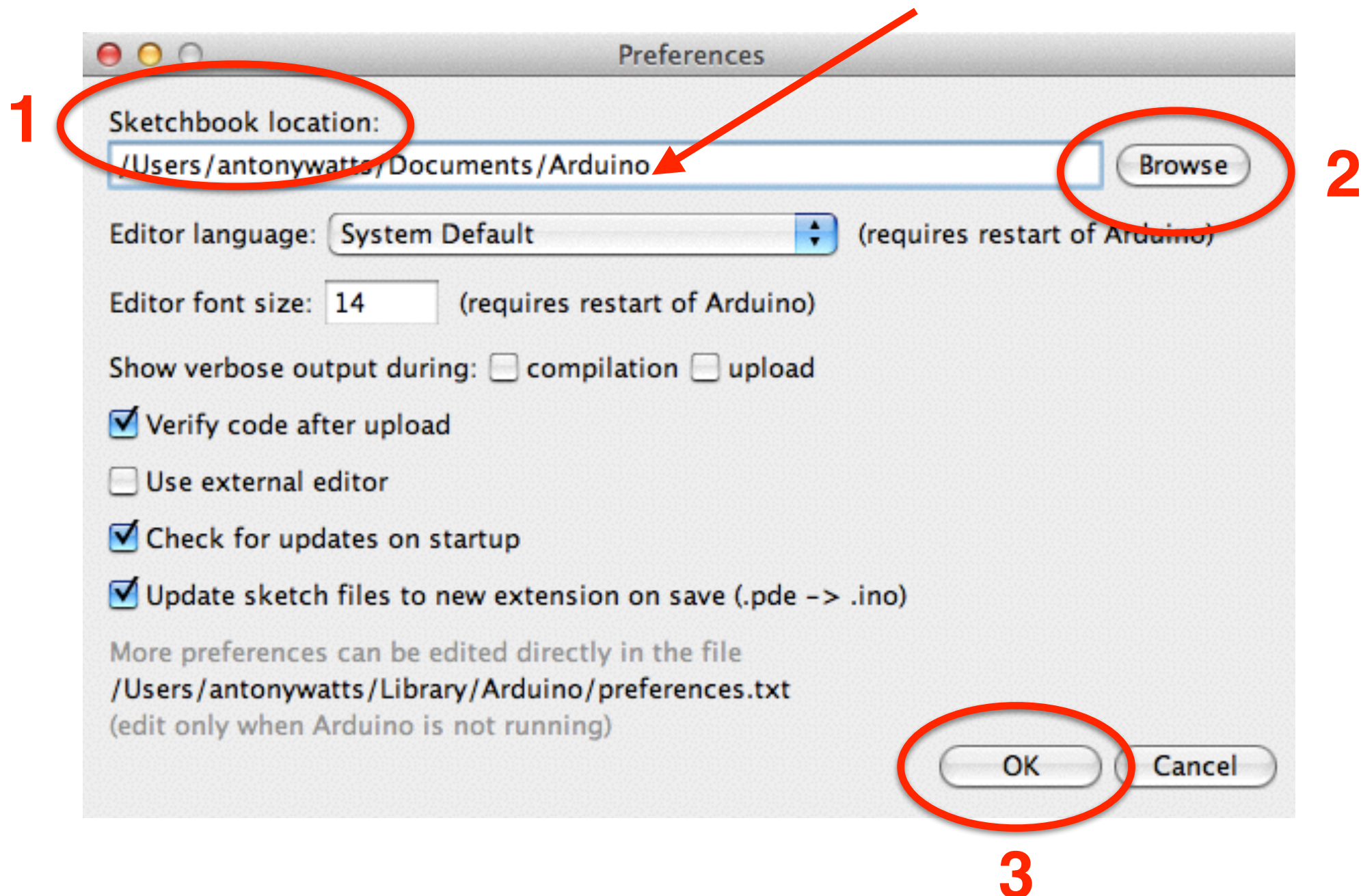
Set up your PC

- Club WiFi: Edimax AP
- Password: 1234567890
- Download and install Arduino IDE from arduino.cc
- Previous course attendees: download the latest version!!!

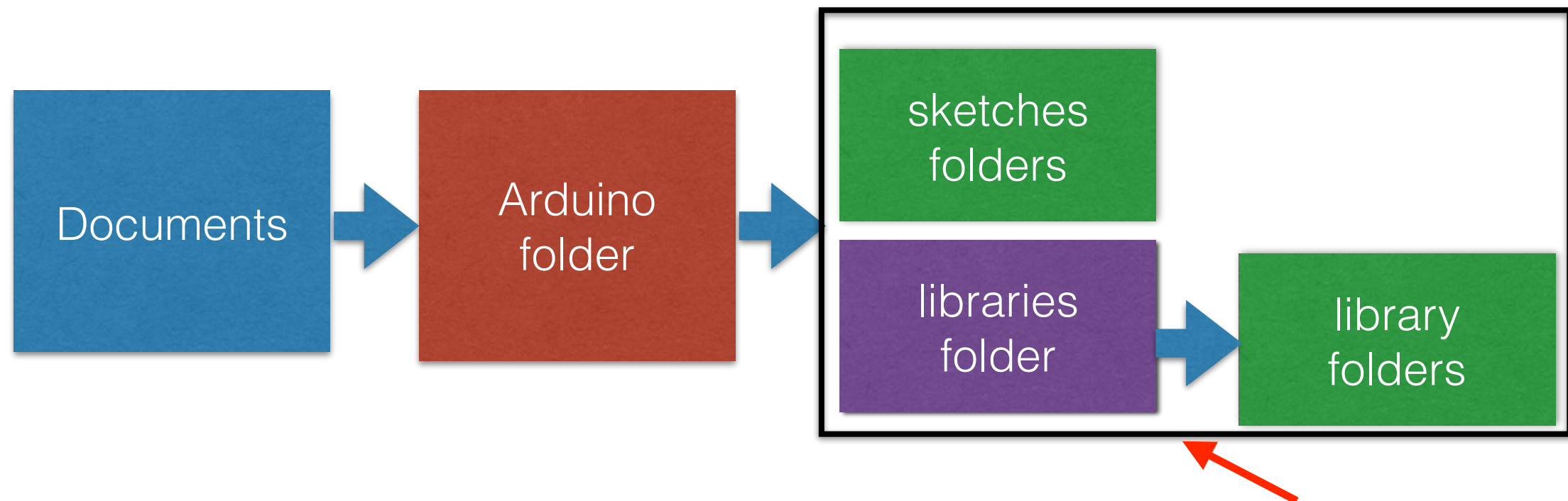


Sketchbook location

- Create a folder, Documents / Arduino
- Start the Arduino program
- Go to Preferences, select the Arduino folder



Copy the USB stick



- Copy the USB stick to your Arduino folder
 - Sketches
 - Libraries folder and libraries
 - HELP and DOCS

USB stick

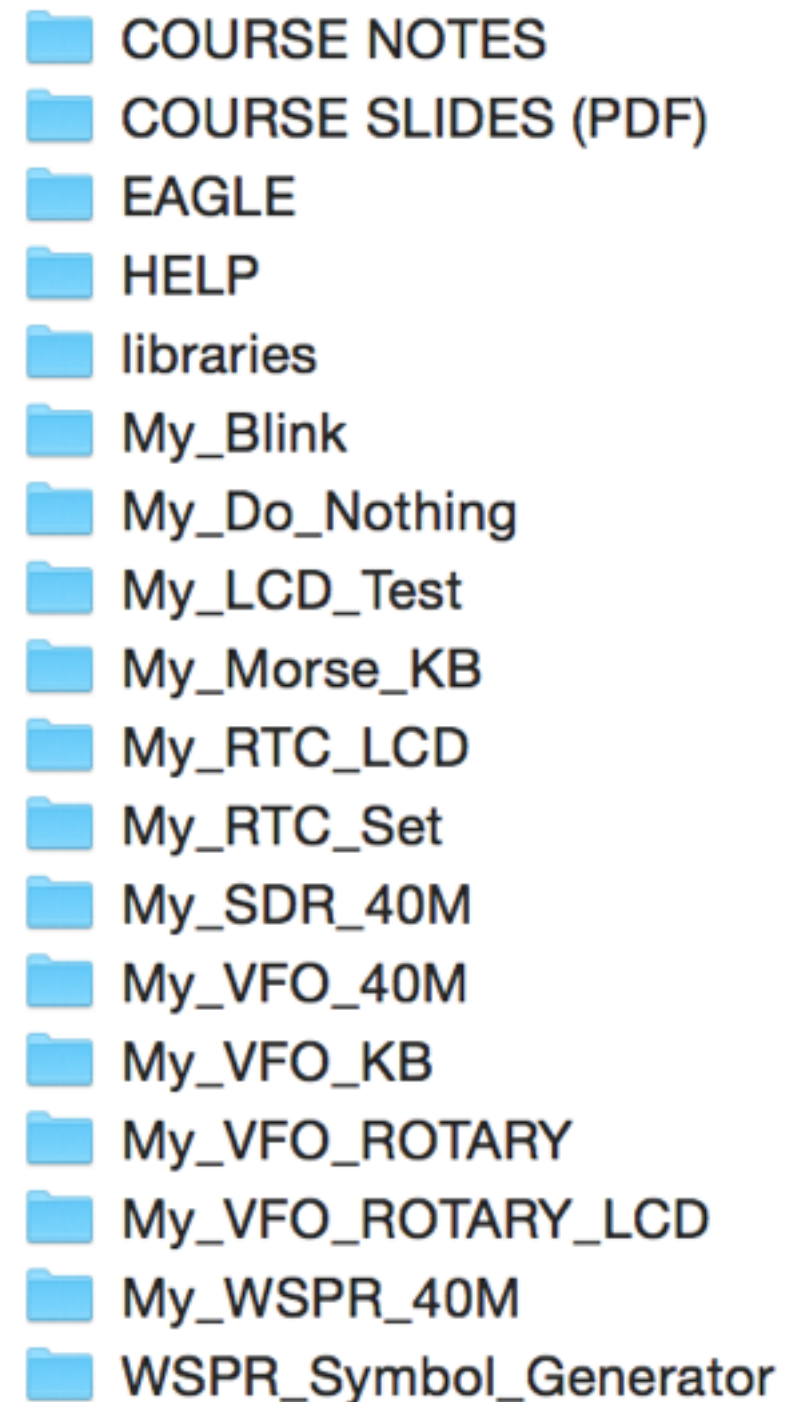


Arduino folder

- Your Arduino folder should look like this:
 - Course notes and slides
 - Eagle PCB design example
 - Various HELP files
 - Libraries
 - Software sketches

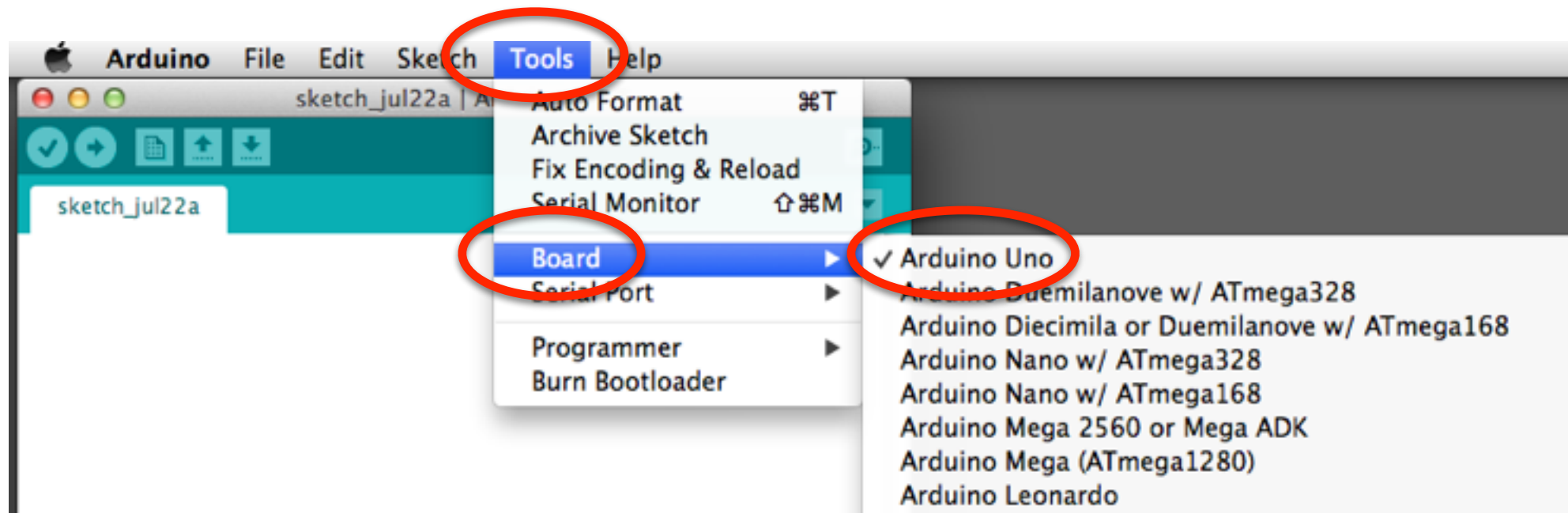


Re-start
Arduino

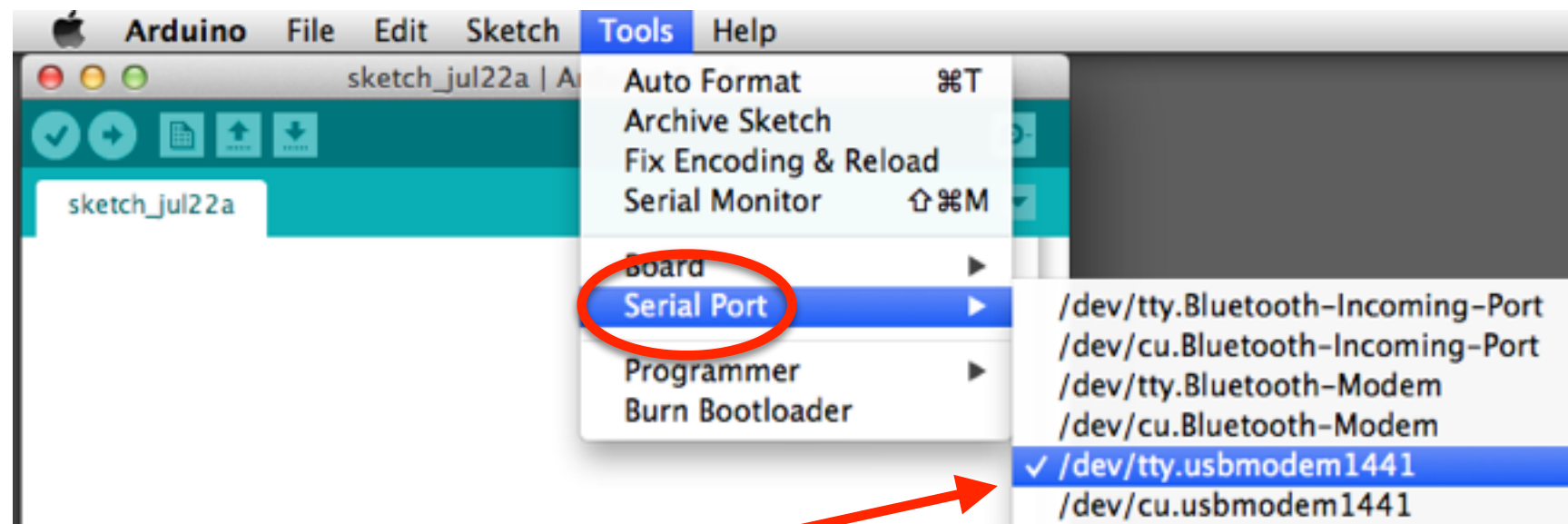


Plug in Board

Board



Port



Chose your
COM port



During this course you will learn to code

Write

Compile

Upload

Source code -> Compiler -> Arduino UNO machine code binary

Software is the key

- Arduino programs are called Sketches
- The key to understanding and using Arduino is learning to code sketches
- Coding is done in a language called 'C'.
 - C is based on functions. `int myFunction(int a, int b){ //do this }`
- Human readable statements are typed into the Arduino IDE program editor
- Arduino IDE then compiles them to computer code and uploads them to your board



Sketches have a basic outline

- File > Sketchbook > My_Do_Nothing
- Sketches have a simple, basic structure
 - “**include**”
 - “**define**”
 - “**setup**”
 - “**loop**”
 - “**myFunction**”
- We will see many sketches all with similar outlines



```
// My_Do_Nothing is a sketch outline, the sketch does
nothing

// example of an included library
→ #include <Arduino.h>

// example of a constant define
→ #define LED 13

// setup function, executed once on upload (->)
→ void setup() {

}

// loop function, executed over and over
→ void loop() {

    c = myFunction(a, b); // call a user function

}

// your own function, called by loop()
→ int myFunction(int a, int b) {

}
```

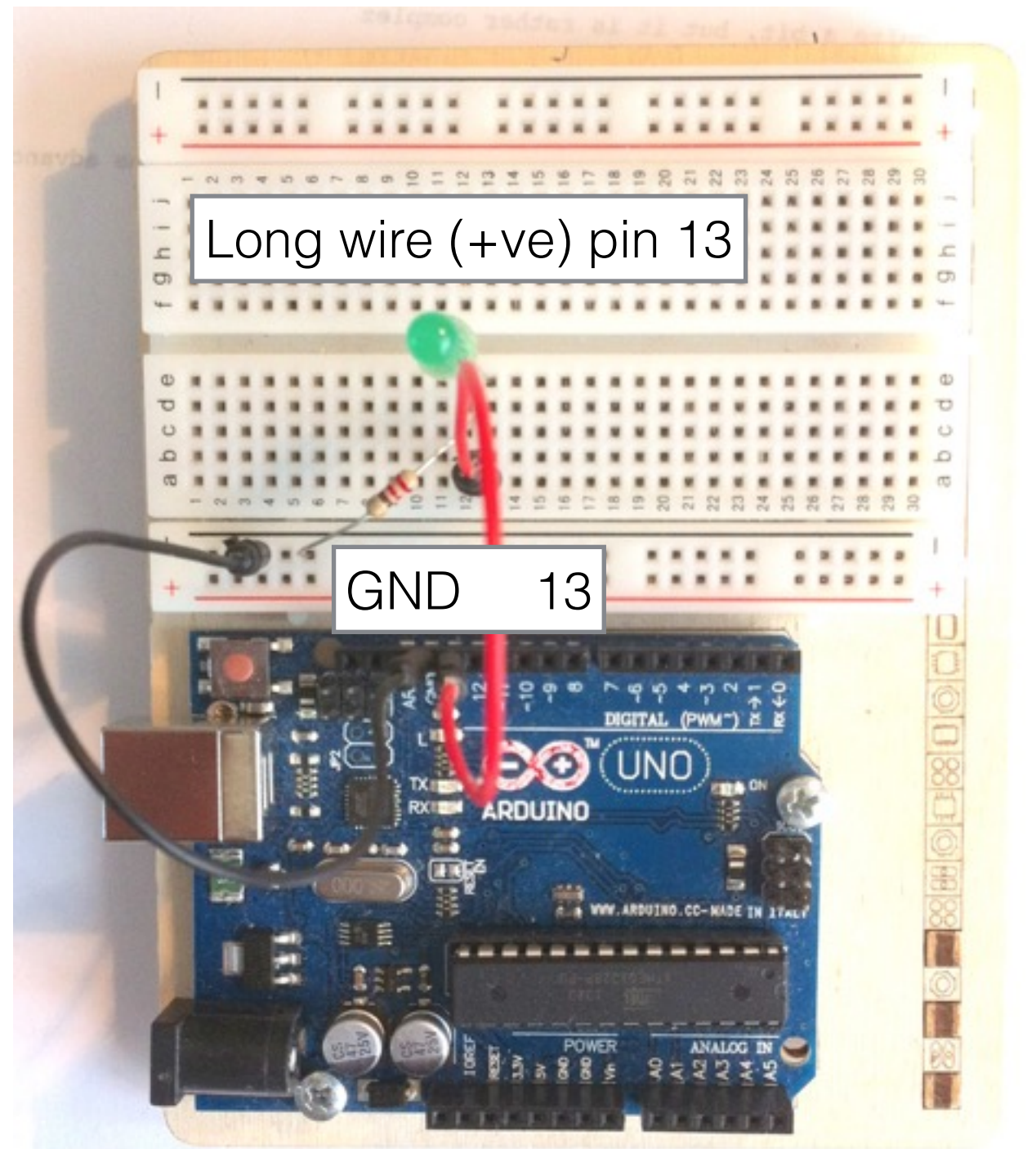
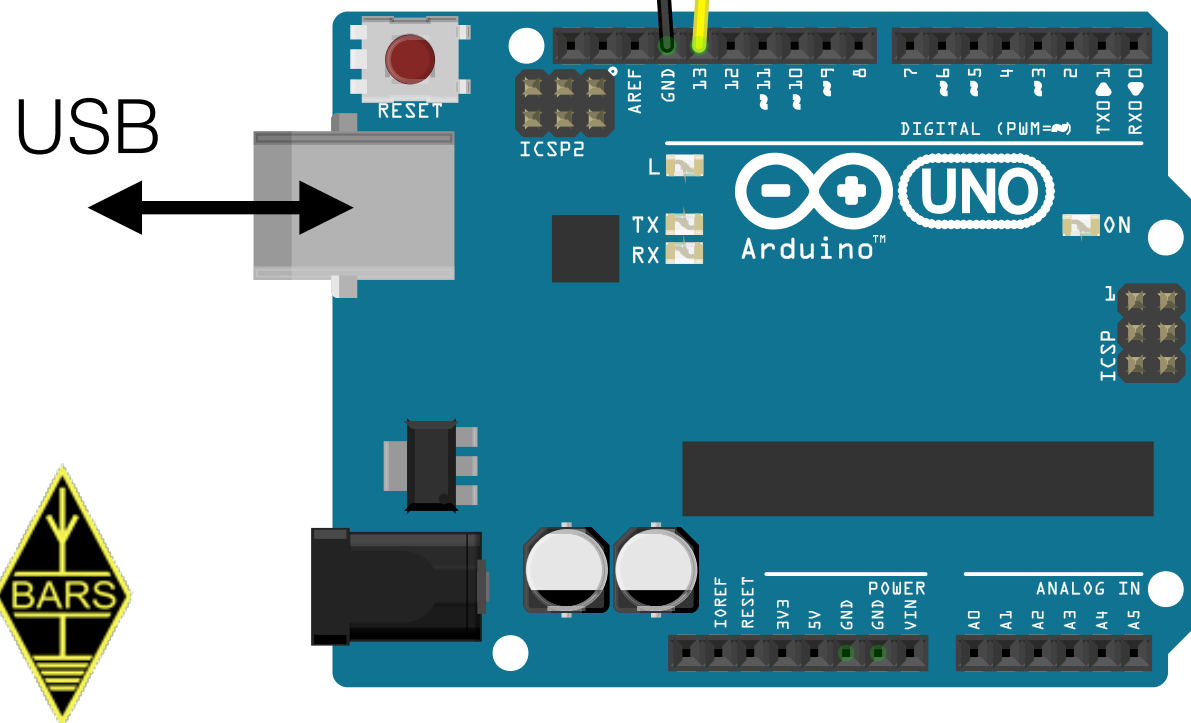
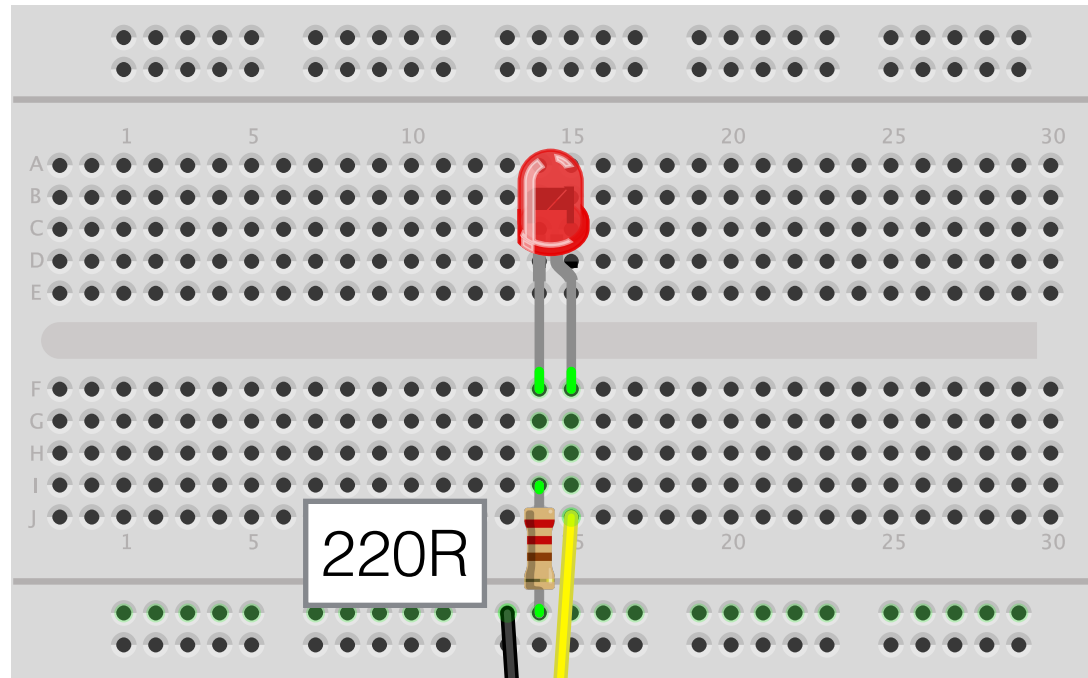
Let's run some

Learner sketches

Blink an LED

Send Morse Code

Wire up



My_Blink

- File > Sketchbook > My_Blink
- Click the Upload button
- The LED connected to pin 13 will blink
- Let's have look at the sketch code and understand it
- Make the LED flash faster...



Upload



My_Blink

```
// My_Blink
// flashes a LED on pin 13

// pin number
→ #define LED 13

// the setup routine runs once when you upload (->) the sketch
→ void setup() {
    // initialise the digital pin 13 as an output
    pinMode(LED, OUTPUT);
}

// the loop runs over and over again, forever
→ void loop() {
    digitalWrite(LED, HIGH); // turn the LED on (HIGH voltage level)
    delay(1000);              // wait for 1 second (1000ms)
    digitalWrite(LED, LOW);  // turn the LED off (LOW voltage leve)
    delay(1000);             // wait for 1 second
}
```

Change to 100 to flash faster



Send Morse

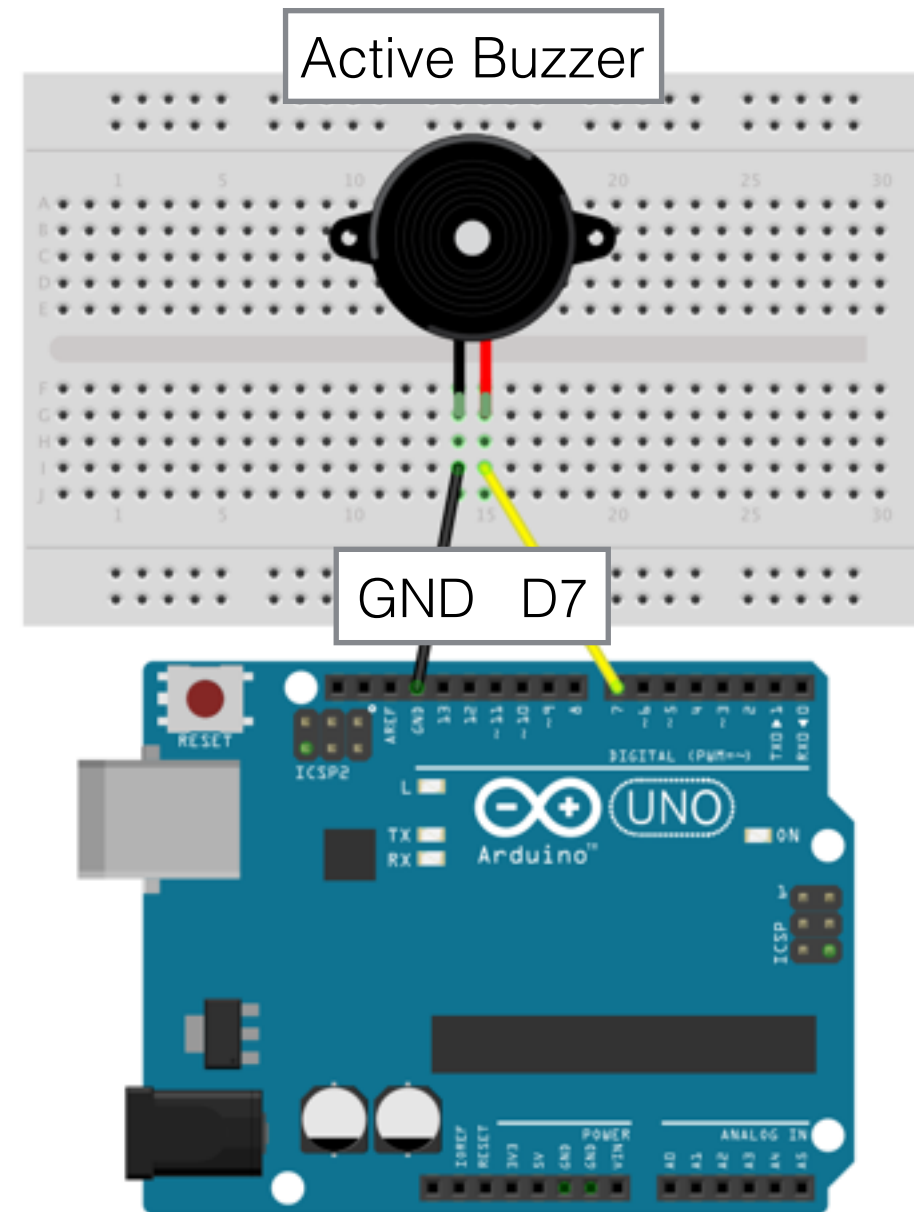
- Wire up the buzzer (GND - D7)
- File > Sketchbook > My_Morse_KB
- Upload
- Open Monitor, enter text message
e.g. "CQ DE <Your Call Sign> K"
- Challenge the guy next to you to read
a your message!
- Buzzer could be replaced by key relay



Arduino
D7



TX Key input



Home work

- Have a look through the code for the My_Morse_KB sketch
- Read it line by line and understand what each line of code does



```
My_Morse_KB | Arduino 1.6.5

1 // My_Morse_KB is a morse morse sender with KB input
2 // active piezo buzzer on pin 7
3
4 // include the special llibrary to generate morse dit/dah
5 #include "MorseEnDecoder.h"
6
7 // define a constant for words per minute
8 #define WPM 5
9
10 // define a constant for the buzzer pin
11 #define BUZZER 7
12
13 // create a morseOut object
14 morseEncoder morseOut(BUZZER);
15
16 // setup runs once on upload
17 void setup() {
18     // set BUZZER pin as an output
19     pinMode(BUZZER, OUTPUT);
20
21     // start serial comms with Arduino IDE monitor window, over USB
22     Serial.begin(9600);
23     while (!Serial); // wait for USB connection
24 }
```

What is

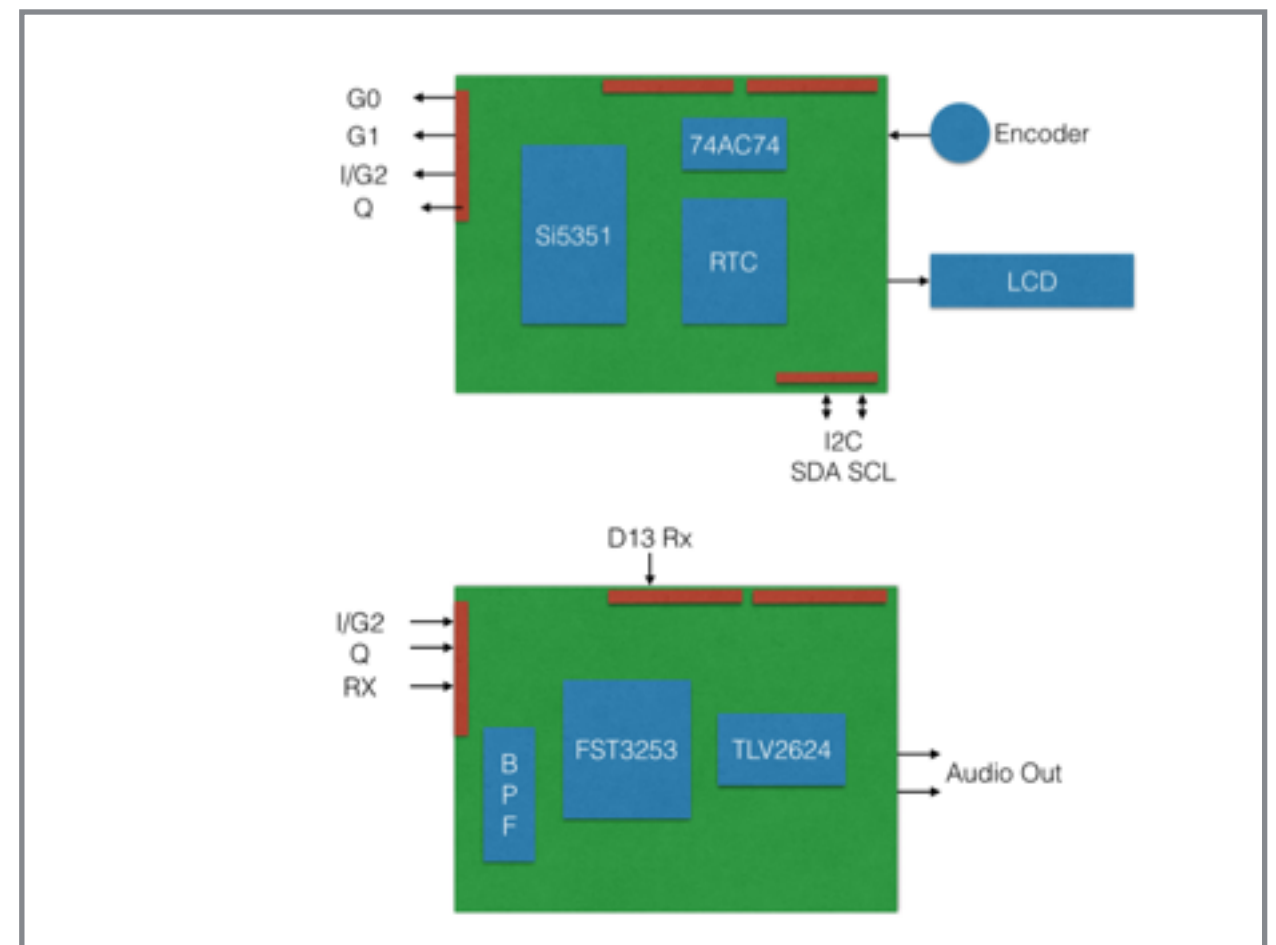
CONCEPT?

Exciting, learning, sharing

Concept plan

Focus on HF

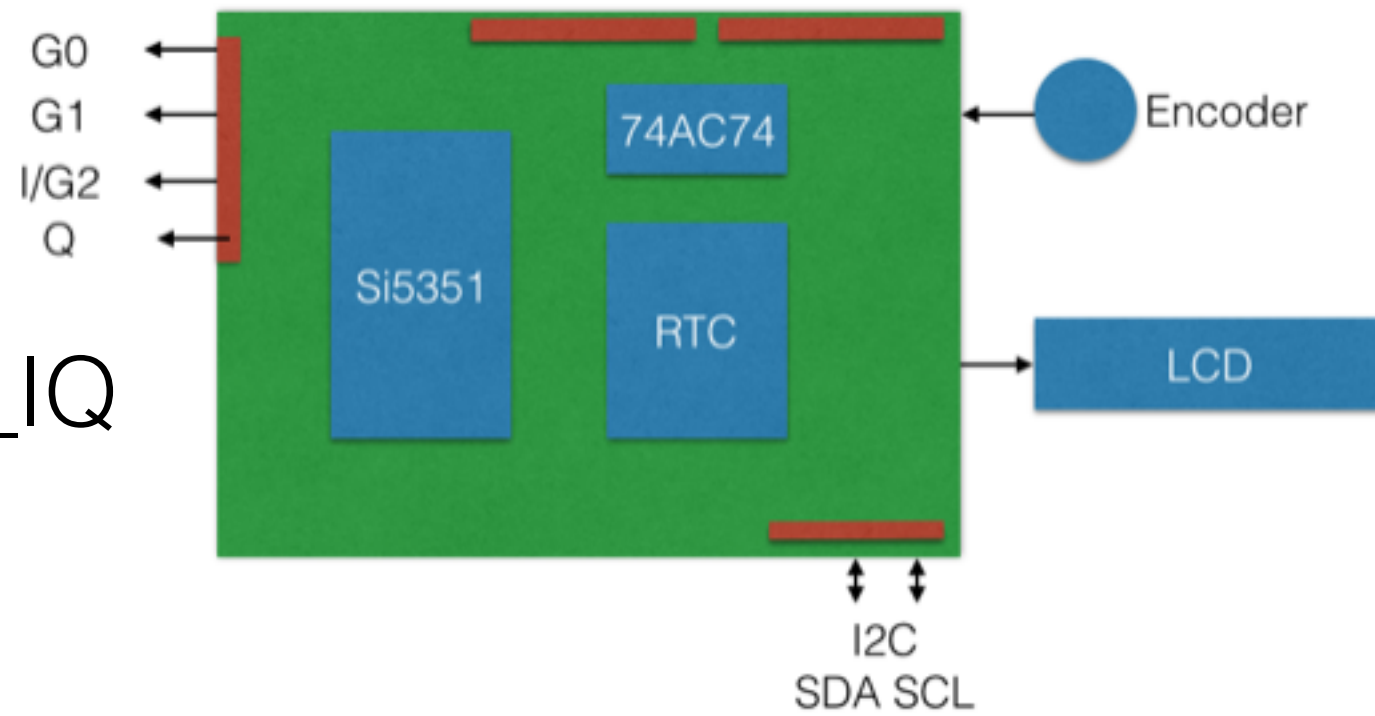
- Develop and build shields that plug into your Arduino
- Start with two shields
 - VFO with RTC
 - 40m SDR RX
- Sketches for modes: VFO, SDR, CW, SSB, QRSS, WSPR - plus others later



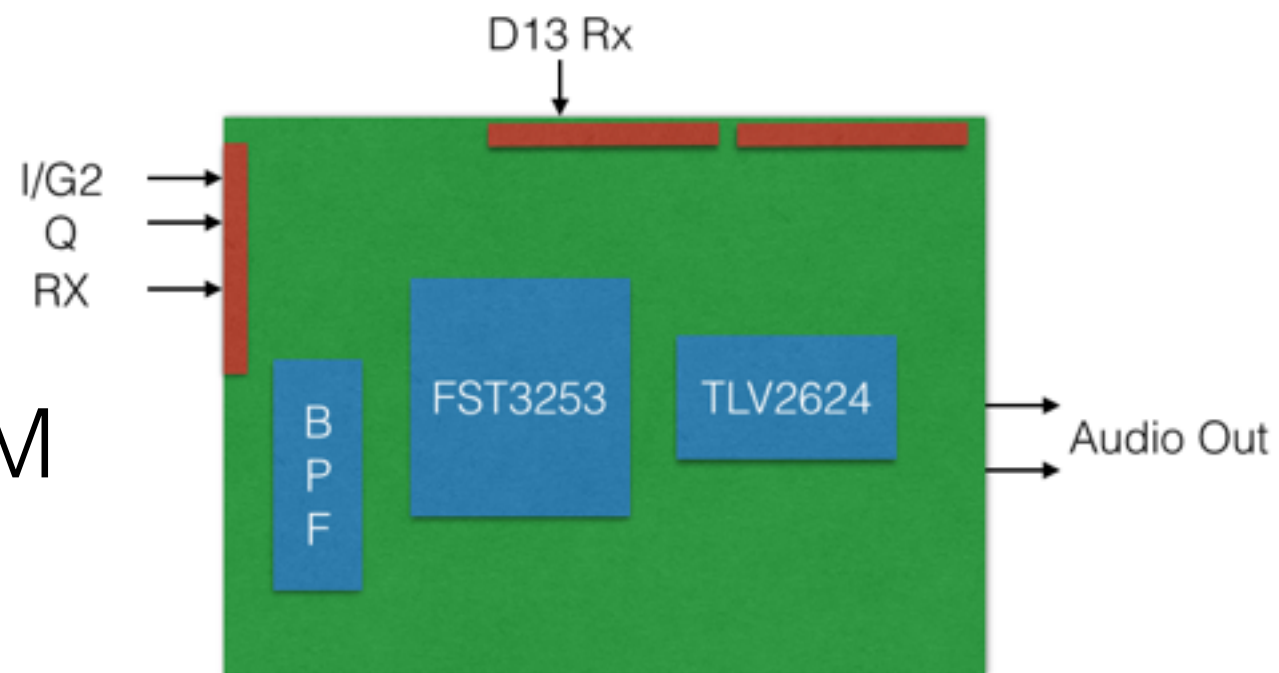
Plus, later
LPF, SDR TX, PA



VFO_RTC_IQ



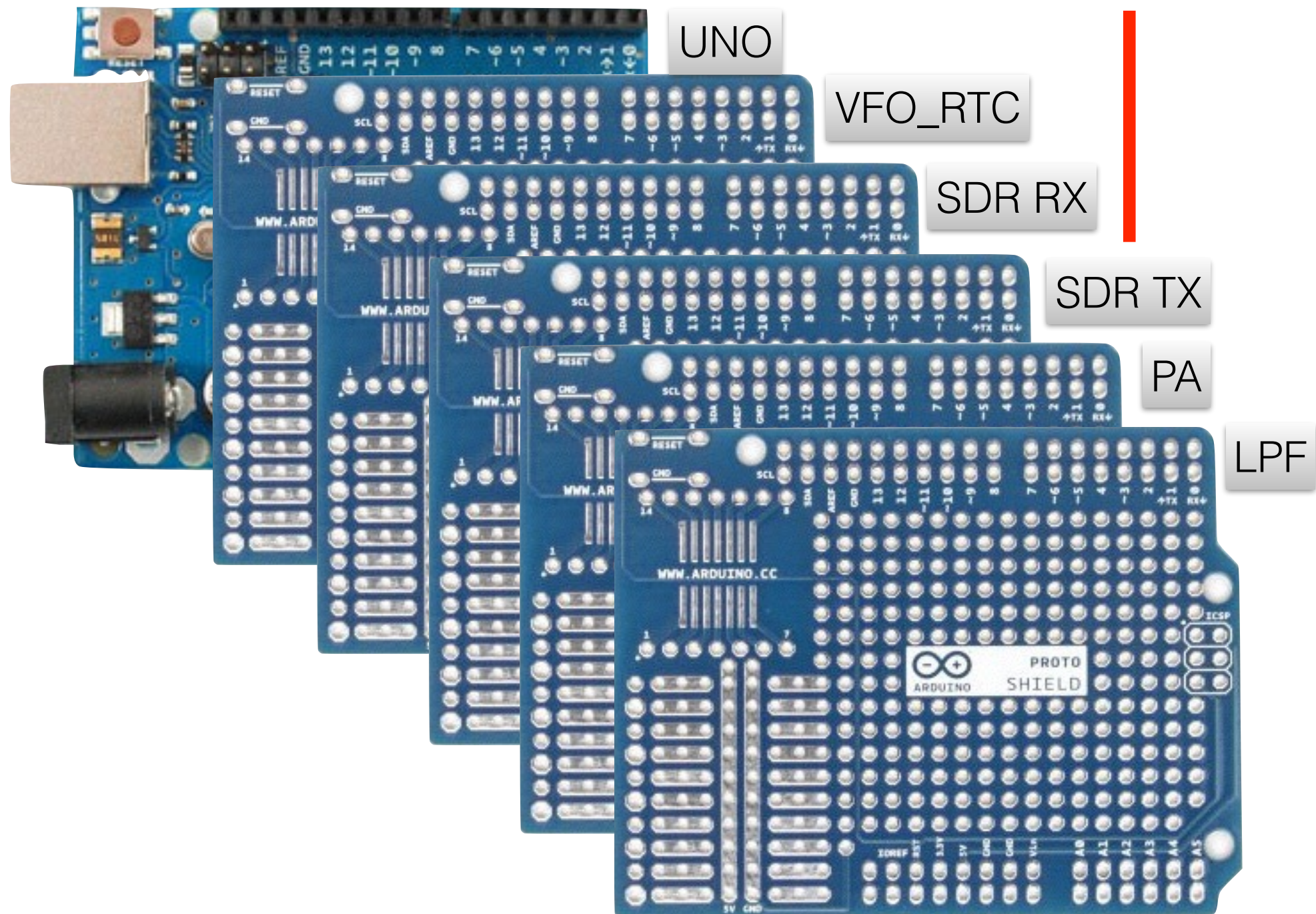
SDR_40M



Shields vs Modes

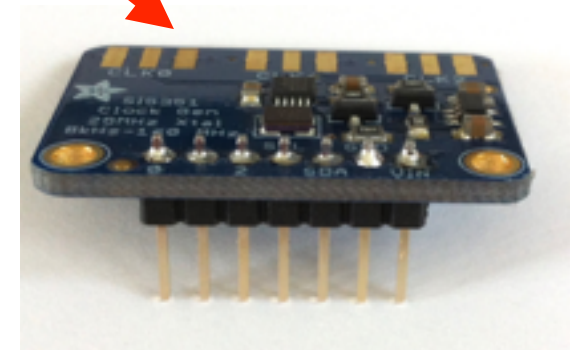
	WSPR	QRSS	CW	SDRX	SDTX	DC RX
UNO	x	x	x	x	x	x
*RTC	x			[x]		
*VFO	x	x	x	x	x	x
*SDRX				x		
SDTX					x	
DC RX						x
PA	x	x	x		x	
LPF	x	x	x	x	x	x

CONCEPT



Being practical

- Later on you will be building the VFO & SDR shields, these carry SMDs
- Check you have a suitable soldering iron. Wire cutters. A small set of tweezers, a magnifier would be useful
- Hobby components has a very good soldering iron (40W temp controlled). Amazon has Solder Flux Pen + 0.3mm Solder and magnifiers.
- **HOMEWORK: Before S2 Solder the header onto the Kit 2 VFO module**



Next time

The serious stuff
VFO and RTC modules