Introduction to robotics

4th lab

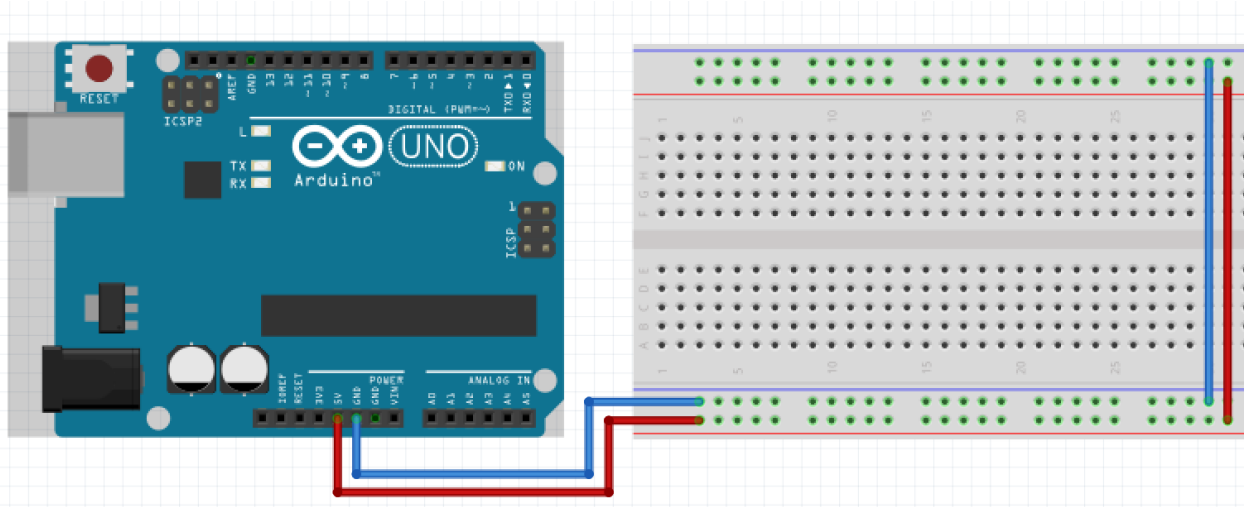
Remember, when possible, choose the wire color accordingly:

- **BLACK** (or dark colors) for **GND**
- **RED** (or colored) for **POWER (3.3V / 5V / VIN)**
- **Remember** than when you use digitalWrite or analogWrite, you actually send power over the PIN, so you can use the same color as for **POWER**
- **Bright Colored** for read signal
- We know it is not always possible to respect this due to lack of wires, but first rule is **NOT USE BLACK FOR POWER OR RED FOR GND!**

Now, let's pick it up where we left off…

Pull out your Arduino and breadboard and connect them like in the schematic. This is to "power up" the breadboard so we can easily have access to **5V** and **GND**.

Attention! Remember how the breadboard works. Use correct wire colors.



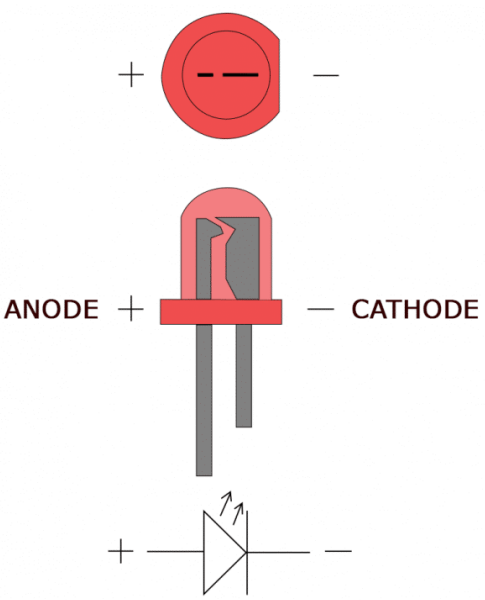
1. 7-segment display

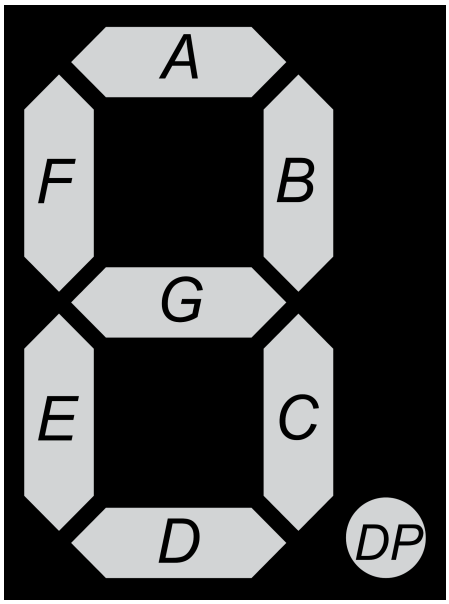
1.1 Recap

Questions:

1. **What is actually a RGB LED?**
1. **So, what's a 7-segment display?**

Remember the LED basics:



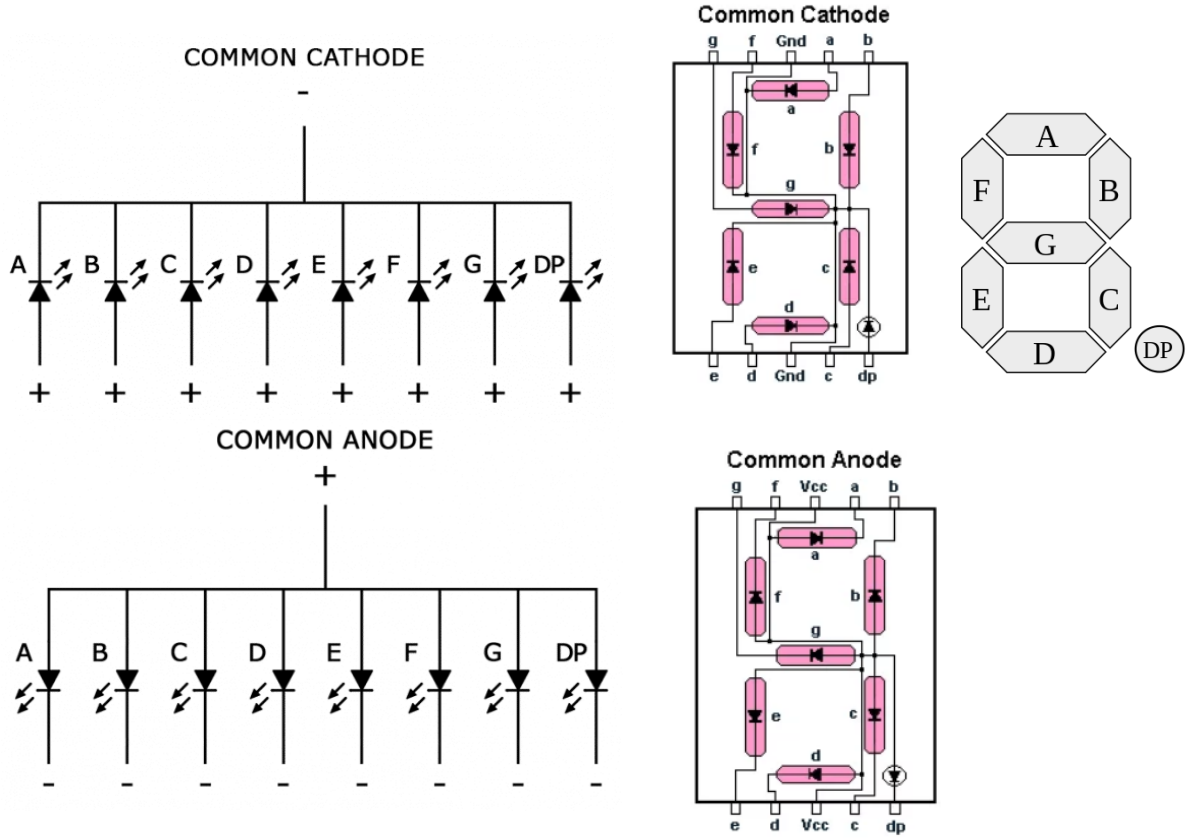


(source: https://en.wikipedia.org/wiki/Seven-segment_display)

1.2 Introduction

The 7-segment display, also written as “seven segment display”, consists of seven LEDs (hence its name) arranged in a rectangular fashion as shown. Each of the seven LEDs is called a segment because when illuminated the segment forms part of a numerical digit (both Decimal and Hex) to be displayed. An additional 8th LED is sometimes used within the same package thus allowing the indication of a decimal point (DP) when two or more 7-segment displays are connected together to display numbers greater than ten.

Like the RGB LED, 7-segment displays can have a **common cathode** or **common anode**. Although you most likely have a common cathode, it’s best to check it with a multimeter.



1.3 Displaying characters

The display system is mainly used for numbers and not letters. That is why some letters are not compatible and are easily mistaken for a number. Short messages giving status information (e.g. "no dISC" on a CD player) are also commonly represented on 7-segment displays. In the

Punctuation encodings

Glyph	Display	Name(s)
-		Minus and Hyphen
=	*	Equals
_		Underscore
°		Degree
[		Left square bracket
]		Right square bracket
		Space

case of such messages it is not necessary for every letter to be unambiguous, merely for the words as a whole to be readable

Hexadecimal encodings for displaying the digits 0 to F[11][12]

Digit	Display	gfedcba	abcdefg	a	b	c	d	e	f	g
0		0x3F	0x7E	on	on	on	on	on	on	off
1		0x06	0x30	off	on	on	off	off	off	off
2		0x5B	0x6D	on	on	off	on	on	off	on
3		0x4F	0x79	on	on	on	on	off	off	on
4		0x66	0x33	off	on	on	off	off	on	on
5		0x6D	0x5B	on	off	on	on	off	on	on
6		0x7D	0x5F	on	off	on	on	on	on	on
7		0x07	0x70	on	on	on	off	off	off	off
8		0x7F	0x7F	on	on	on	on	on	on	on
9		0x6F	0x7B	on	on	on	on	off	on	on
A		0x77	0x77	on	on	on	off	on	on	on
b		0x7C	0x1F	off	off	on	on	on	on	on
C		0x39	0x4E	on	off	off	on	on	on	off
d		0x5E	0x3D	off	on	on	on	on	off	on
E		0x79	0x4F	on	off	off	on	on	on	on
F		0x71	0x47	on	off	off	off	on	on	on

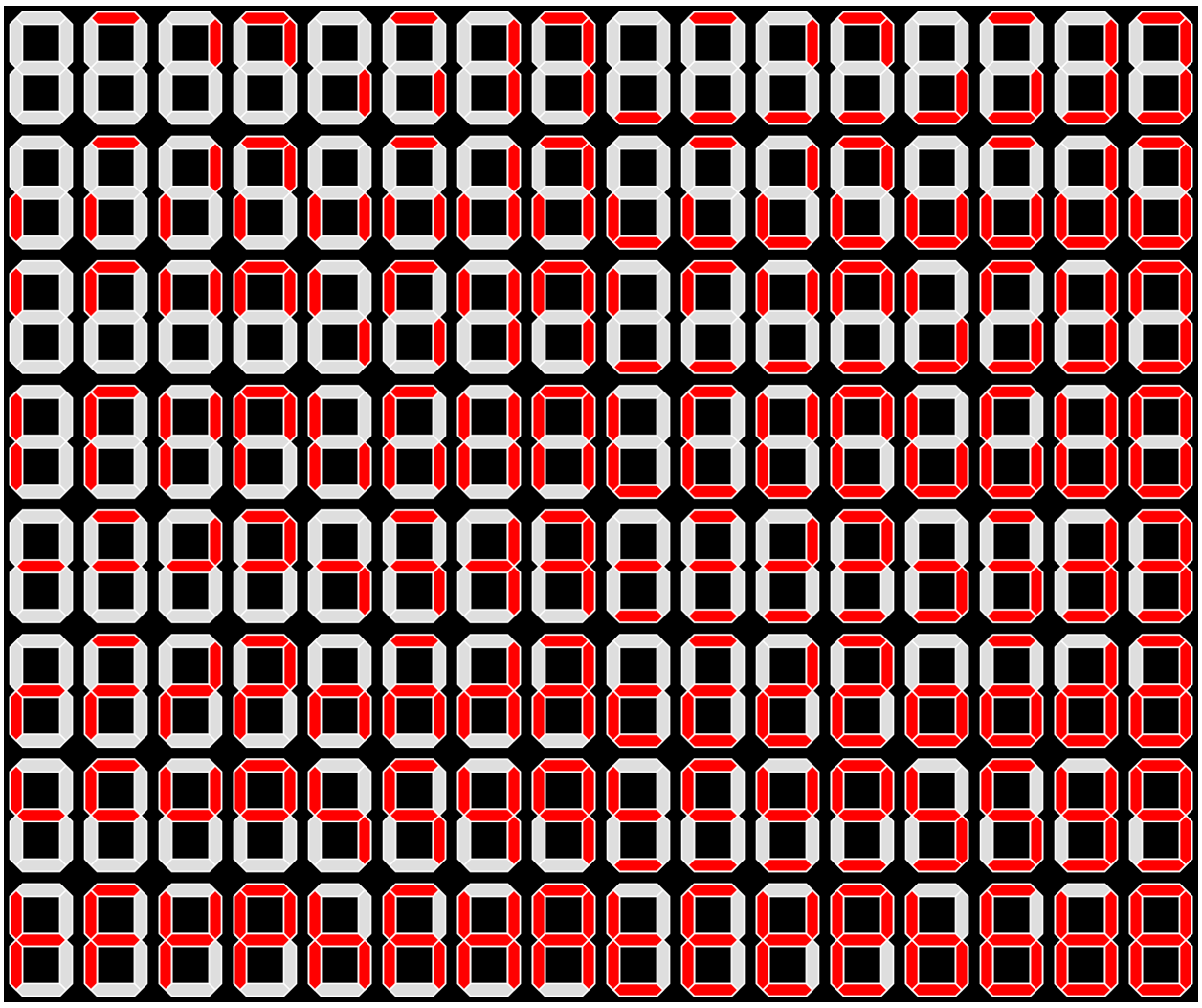
(source: https://en.wikipedia.org/wiki/Seven-segment_display).

Example of short messages

OPEn, CLOSE, PLAY, PAUSE, SHUFFLE, no dISC

StArt, StoP, run, FAIL, Error, SEtUP, HELP

on, oFF, YES, no, Hot, CoLd



16x8-grid showing the 128 states of a seven-segment display

To determine which type of display you have, probe the pins with a multimeter. Try connecting the middle pin to the **COM - black -** of the multimeter, and any other pin (except the middle one on the other side) to the **V+ - red - .** If any of the segments start lighting, it's a common cathode. If not, try reversing the wires; if any of the segments start lighting then it's a common anode.

Questions:

1. What type of display do you have?

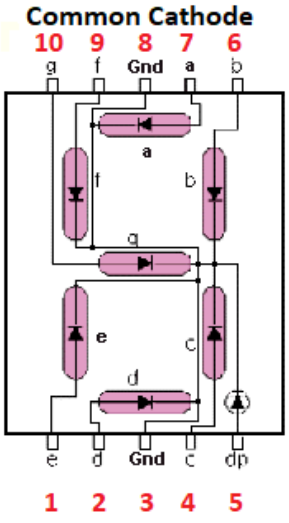
1.4 Connecting the 7-segment display

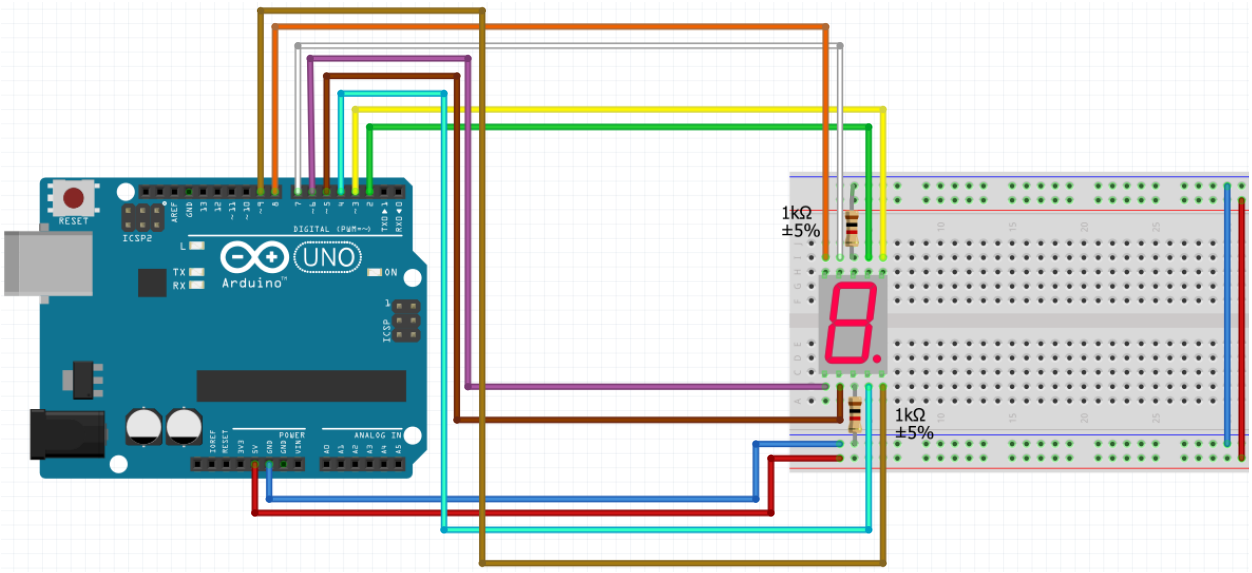
First of all, check which type of display you have. Common cathode must have Ax at the end. (x can have any value). If you don't have a common cathode, ask for another one.

Common cathode: (5161Ax)

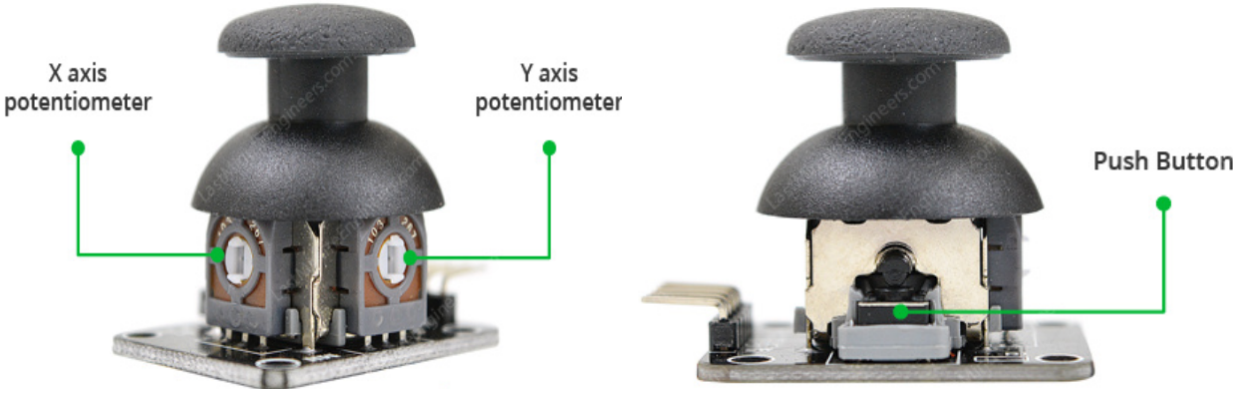
Connecting the 7 segment display to the Arduino board:

Attention! The pins have a corresponding letter and a number based on their circular order, starting with 1 from the bottom left. The red numbers are **NOT** the corresponding Arduino pins.

Display PIN	Arduino PIN	Schematic
A (7)	2	
B (6)	3	
C (4)	4	
D (2)	5	
E (1)	6	
F (9)	7	
G (10)	8	
DP (5)	9	

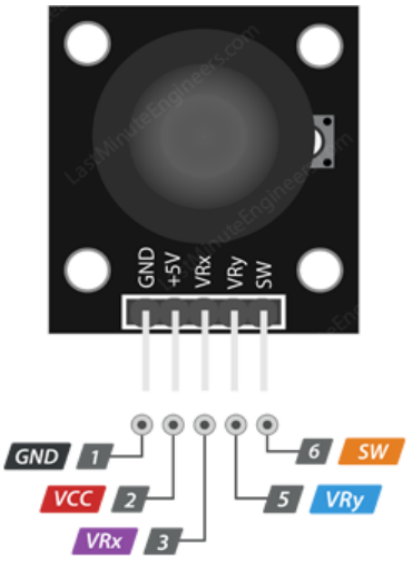


2. Joystick



The joystick is similar to two potentiometers connected together, one for the vertical movement (Y-axis) and other for the horizontal movement (X-axis). It also contains a switch which activates when you push down on the cap.
What we're gonna use in this lab is a self-centering spring loaded joystick, meaning when you release the joystick it will center itself. It also contains a comfortable cup-type knob/cap which gives the feel of a thumb-stick.

2.1 Pinout



GND is the Ground Pin
VCC supplies power for the module
VRx gives readout of the joystick in the horizontal direction (X-coordinate) i.e. how far left or right the joystick is pushed.
VRy gives readout of the joystick in the vertical direction (Y-coordinate) i.e. how far up and down the joystick is pushed.
SW is the output from the pushbutton. It's normally open, meaning the digital readout from the SW pin will be HIGH. When the button is pushed, it will connect to GND, giving output LOW.

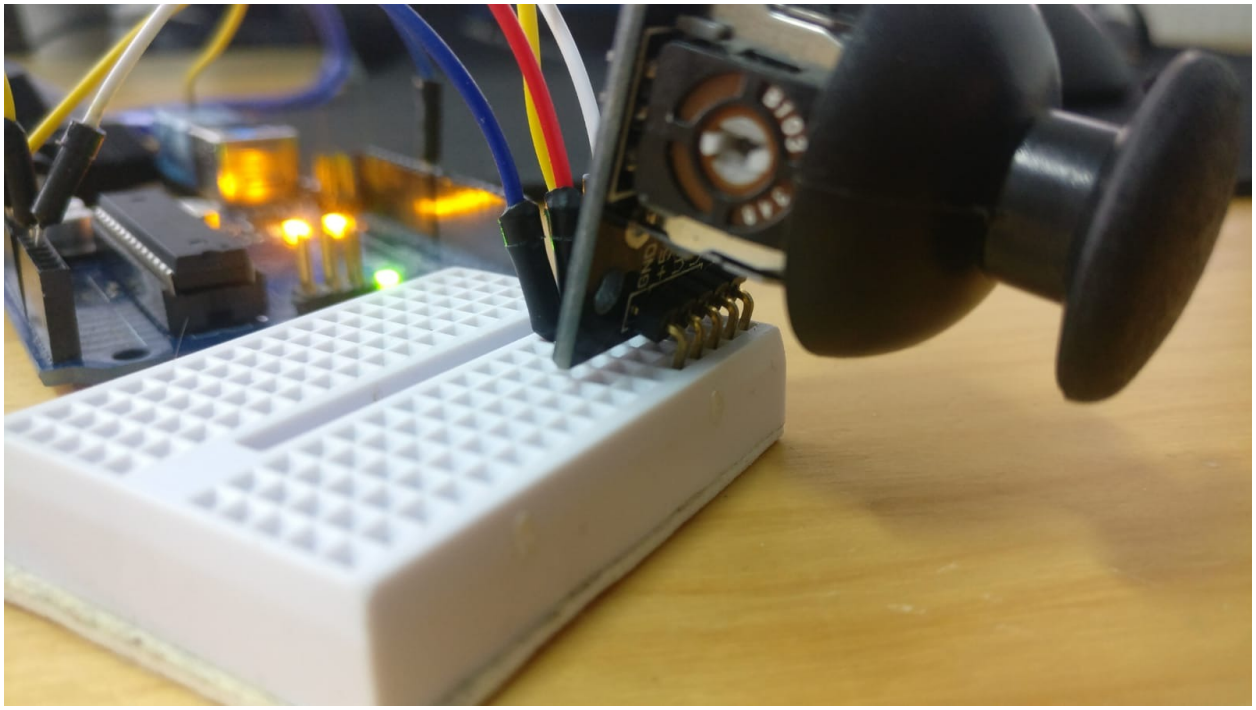
2.2 Reading values from Joystick

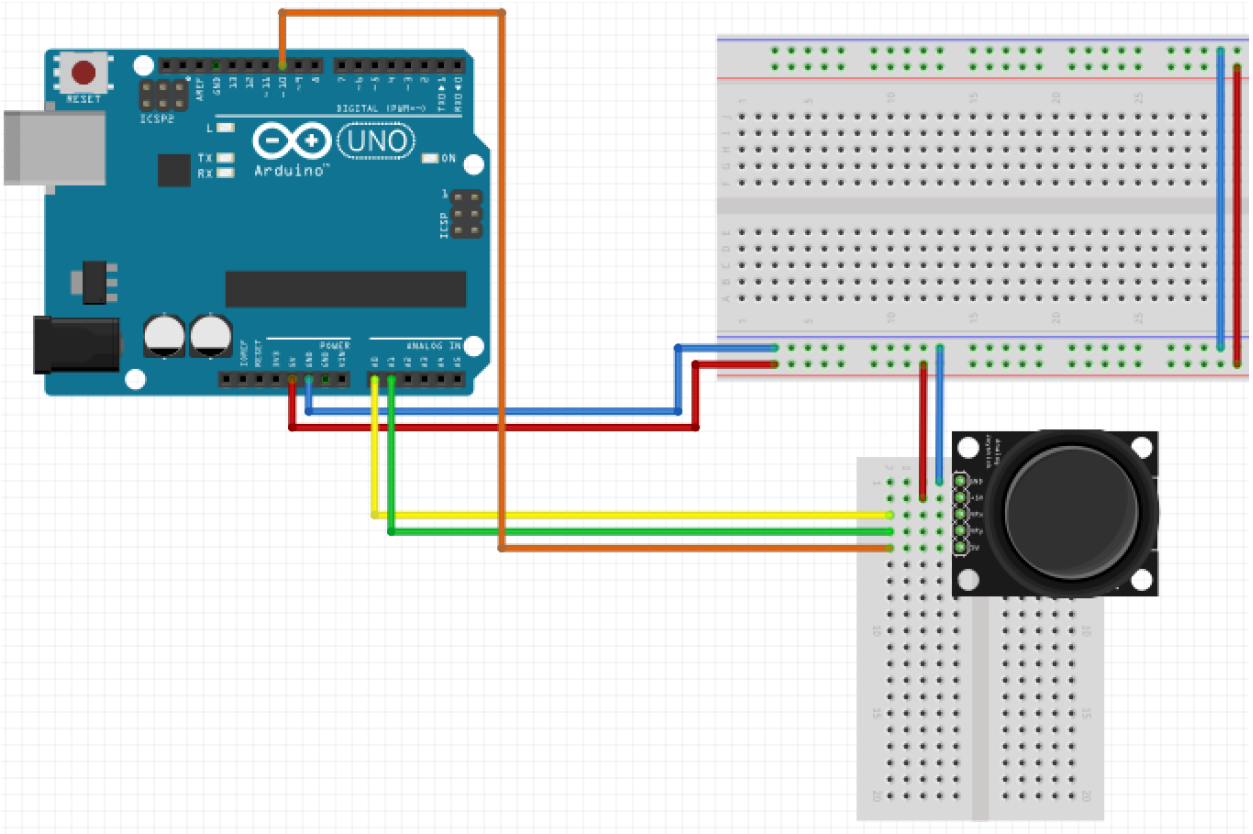
Questions:

1. Now we know we have 2 potentiometers and a pushbutton. How many and what type of pins do we need?
    a. **A**: 2 analog pins for potentiometers and 1 digital pin for pushbutton
2. We have analog data processed by a digital processor (microcontroller). How is this possible?
    a. **A:** By using an intermediate device to convert the analog data into digital data, named ADC (Analog to Digital Converter)

In order to read the joystick's physical position, we need to measure the change in resistance of a potentiometer.

We'll use the joystick with the small breadboard so that we can add the other elements on the medium breadboard without removing the joystick. Make sure you connect your joystick in such a way that you can use it (don't put the wires in front of it).




RESET
ICSP2
UNO
Arduino
TX
RX
ON
ICSP
POWER
ANALOG IN

```
bool digitMatrix[noOfDigits][segSize - 1] = {
// a  b  c  d  e  f  g
  {1, 1, 1, 1, 1, 1, 0}, // 0
  {0, 1, 1, 0, 0, 0, 0}, // 1
  {1, 1, 0, 1, 1, 0, 1}, // 2
  {1, 1, 1, 1, 0, 0, 1}, // 3
  {0, 1, 1, 0, 0, 1, 1}, // 4
  {1, 0, 1, 1, 0, 1, 1}, // 5
  {1, 0, 1, 1, 1, 1, 1}, // 6
  {1, 1, 1, 0, 0, 0, 0}, // 7
  {1, 1, 1, 1, 1, 1, 1}, // 8
  {1, 1, 1, 1, 0, 1, 1}  // 9
};
```