

# Exercise 4.1: shift register

In this exercise you will learn what you can do when you run out of pins on an arduino.

## Hardware Required

Arduino UNO R3

LEDs 8pcs

330 ohm resistors 8pcs

Shift register

0.1 $\mu$ F capacitor

Button

Potentiometer

## Circuit

In this exercise you will get help with the circuit and some of the basic code to make the shift register work.

You can either:

1. Control the leds through serial monitor
2. Use a button to make leds light up the way you want
  - a. Make one successive led light up each time you press the button. When you get to 8 turn all off or turn one off successively.
  - b. Make a bit counter 1 = 1000000, 2 = 01000000, 3 = 11000000, and so on
3. Use a potentiometer to either make a pattern or to increase the "bit counter" the more you turn the potentiometer.

## Code

You will get some basic example code so that you know what to do and how to use the shift register. It's up to you to decide what you want to try. You can find information about the arduino and its libraries on <https://www.arduino.cc/reference/en/> or ask one of the instructors for this course.

## Useful information

If you don't know about binary numbers here's some useful information.

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
128	64	32	16	8	4	2	1

Each number used becomes a one and the rest becomes zeros. You can use a decimal to binary converter like <https://www.rapidtables.com/convert/number/decimal-to-binary.html> if you don't want to calculate the number.

The binary strings sent to the shift register is an 8 bit string so it contains any combination of 8 bits. It is easier in practise, and explained in more detail further down. To light up a LED (make a pin HIGH) we send a one and to turn one off we send a zero.

Binary	Decimal
0000 0001	1
0000 0010	2
0000 0011	3
0000 0100	4

0000 0101	5
0000 0110	6
0000 0111	7
0000 1000	8
0000 1001	9
0000 1010	10
0000 1011	11
0000 1100	12
0000 1101	13
0000 1110	14

When using a shift register there's some basics you need to know. We use 3 pins on the arduino to control the 8 outputs/inputs of the shift register.

Clockpin:

Is used to tell a component like a shift register when a byte has been sent to make sure it is saved. When we send the information the clockpin is low and when a bit has been sent the clockpin should be set to high to confirm that value.

Datapin:

Is used to send the data to the shift register, telling it which pins that should go HIGH when the latchpin goes HIGH.

Latchpin

When this pin is enabled(set to high) the shift register sets ones to HIGH and zeros to LOW for its outputs.

If we want to make it easy for ourself we can use the function `shiftOut()`, more information on:

<https://www.arduino.cc/reference/en/language/functions/advanced-io/shiftout/>

# Exercise 4.2: more shift register

In this exercise you will learn what you can do when you run out of pins on an arduino.

## Hardware Required

Arduino UNO R3

LEDs 8pcs

330 ohm resistors 8pcs

Shift register

0.1 $\mu$ F capacitor

Button

## Circuit

In this exercise you will use the circuit from the last exercise, including a button.

The goal in this exercise is to make a stopwatch. If you want to make it easy for yourself you can make a binary stopwatch using 8 leds. If you want to make it a challenge you can use a 7-segment display.

## Code

You will get some basic example code so that you know what to do and how to use the shift register. It's up to you to decide what you want to try. You can find information about the arduino and its libraries on <https://www.arduino.cc/reference/en/> or ask one of the instructors for this course.

## Useful information

Here's some help to get you started with the stopwatch. You can use the function `millis()` to count the time from when the button is pressed and then you can "push" out the values to the shift register to be displayed on the LEDs or 7-segment.

<https://www.arduino.cc/reference/en/language/functions/time/millis/>