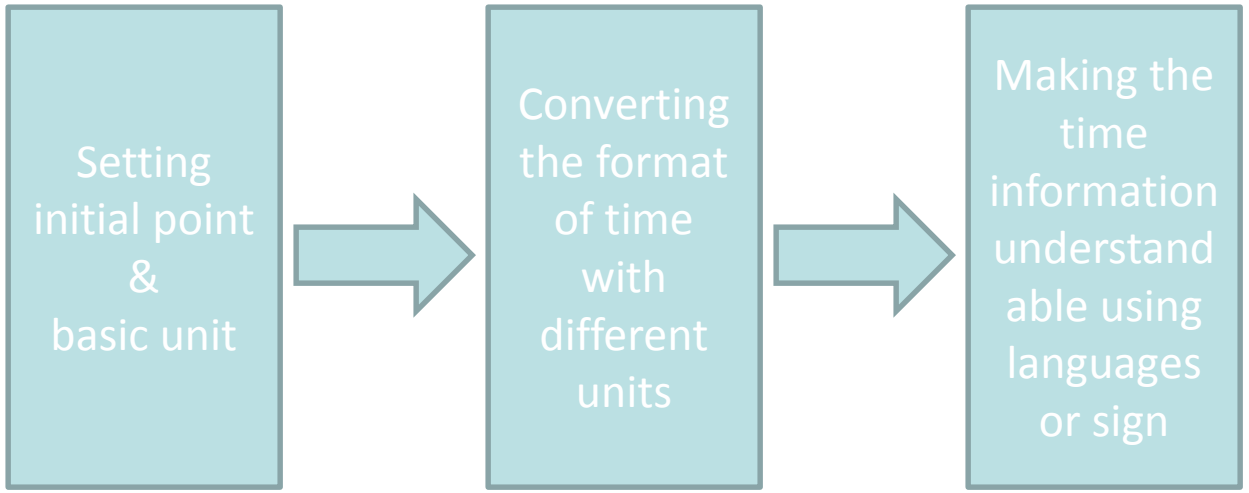
EEE116 Experimental, Computer Skills and Sustainability

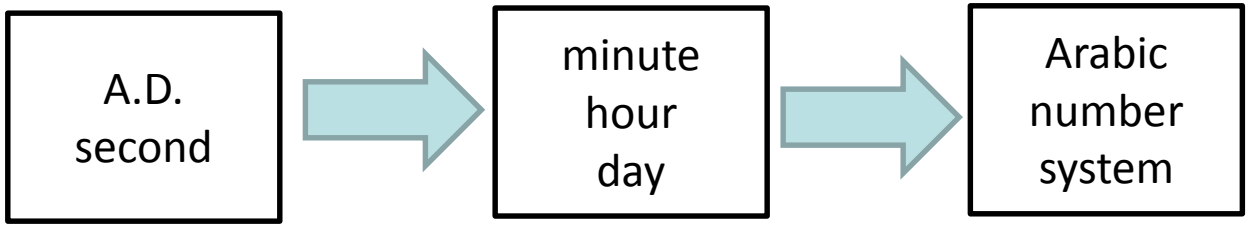
Week 3 Digital clock pre-lab tutorial

Xi'an Jiaotong-Liverpool University

西交利物浦大学

Processing of Time Information





Digital clock project

To design a digital clock which can be started, stopped and reset. It should be able to count up to a certain number, such as, 59 seconds (i.e. from 0,1,2,3....57, 58, 59, 0, 1, 2, 3....) and output the seconds by a pair of 7-segment LED displays.

This is a design project, there is no step by step instruction for you to follow.

Milestones

Functional blocks

To start on the design of your clock, break the whole system down into functional blocks such as the timing generator, the decoder, the display, the set logic, etc.

Complete design

Specify how the functional blocks interface with each other. At this point you can have a completed design with a full schematic in Multisim or online simulation tools. (Optional)

https://circuits.io

Milestones

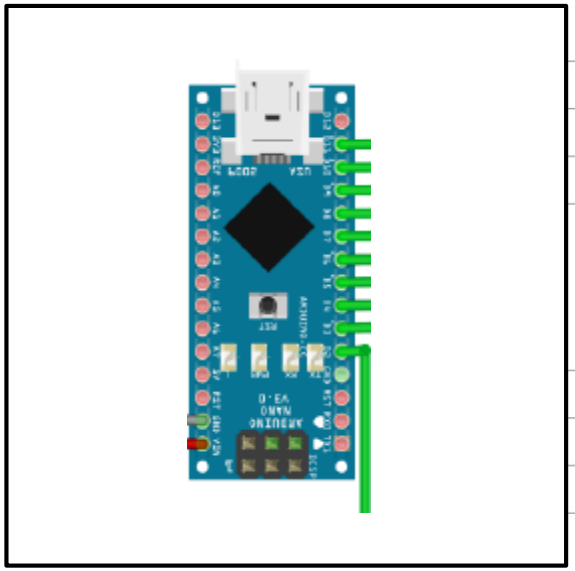
Building and testing

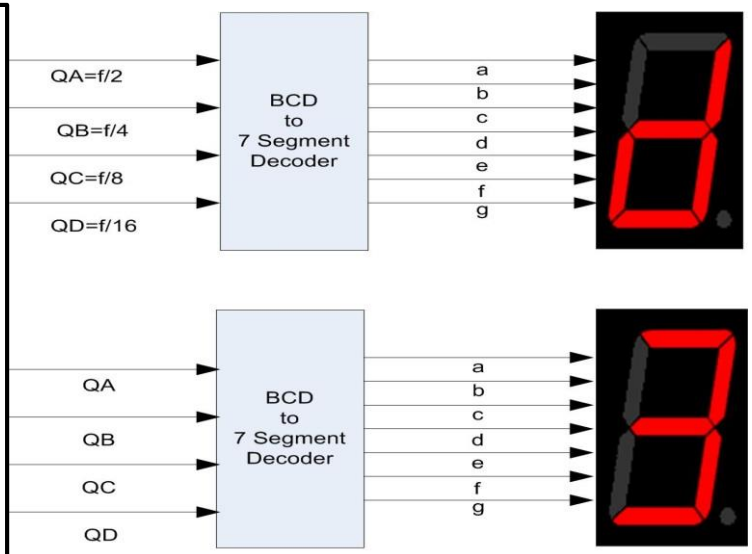
A good approach is to build each of your functional blocks from milestone M1 and test them independently. For example, use a function generator as input signal for a block and look at the output of the block using the oscilloscope.

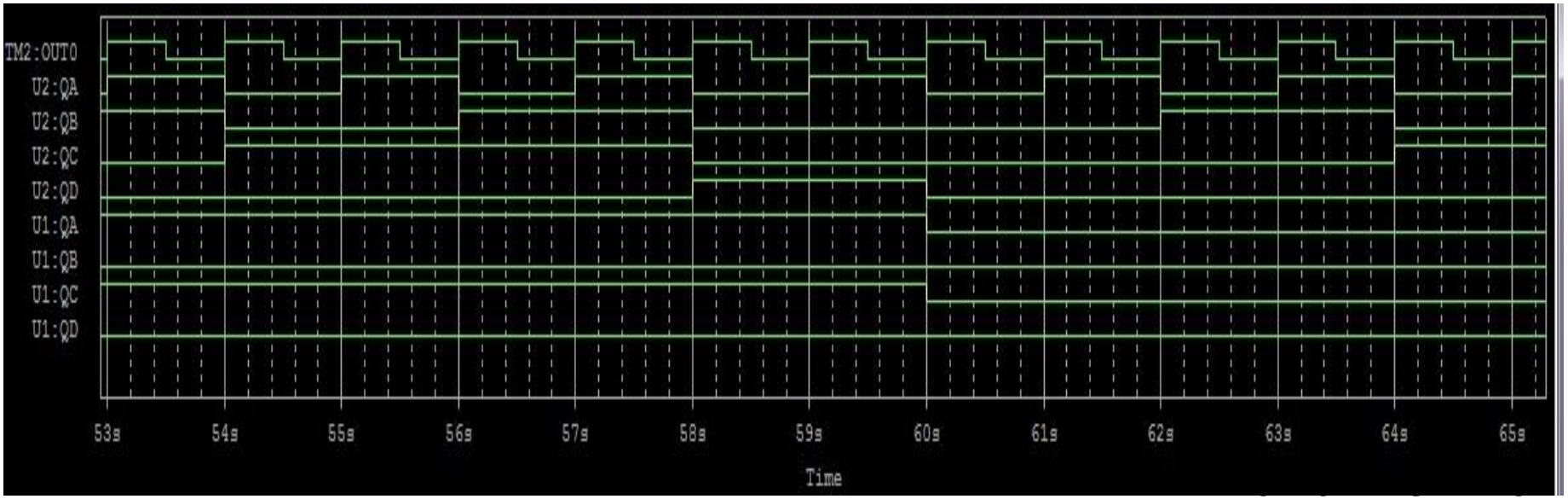
Demonstration of Working Clock.

You must demonstrate your working clock to your TA before the end of the lab. Take a short video clip for your work (as backup).

Basic Structure







Generate time signal from Arduino

Xi'an Jiaotong-Liverpool University 西交利物浦大學

SingleDig_0_to_10_v1 | Arduino 1.8.1

File Edit Sketch Tools Help

SingleDig_0_to_10_v1 §

```
int inputs[4] = {8,9,10,11}; // A,B,C,D inputs
byte BCD[16][4] ={{0,0,0,0},
{1,0,0,0},
{0,1,0,0},
{1,1,0,0},
{0,0,1,0},
{1,0,1,0},
{0,1,1,0},
{1,1,1,0},
{0,0,0,1},
{1,0,0,1},
{0,1,0,1},
{1,1,0,1},
{0,0,1,1},
{1,0,1,1},
{0,1,1,1},
{1,1,1,1}}; //BCD code

void setup() {
for(int a = 0; a < 4; a++){
pinMode(inputs[a], OUTPUT);} //set outputs
}
void loop() {
static int num = 0;
for(int c = 0; c < 4; c++){
digitalWrite(inputs[c], BCD[num][c]);
}
  num++;
  num = num % 10;
  delay(1000);
}
```

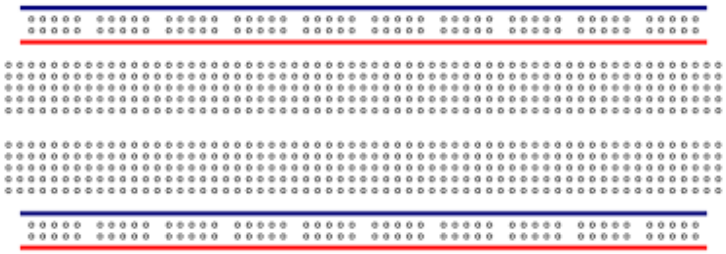
digitalNum | Arduino 1.8.1

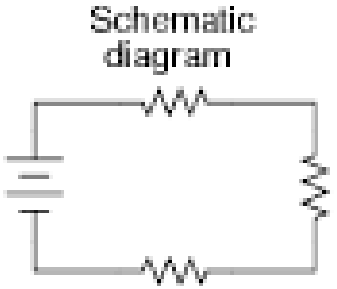
File Edit Sketch Tools Help

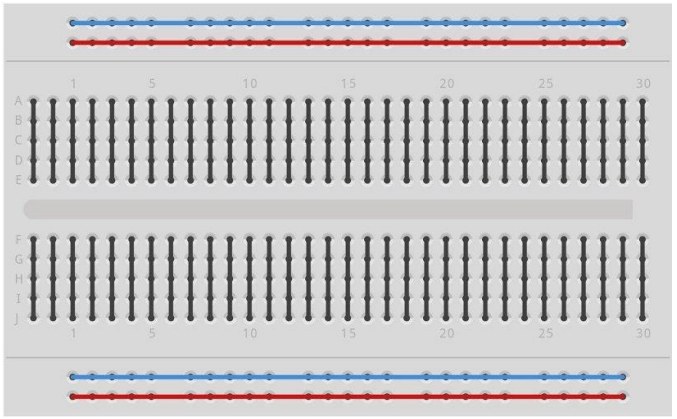
digitalNum

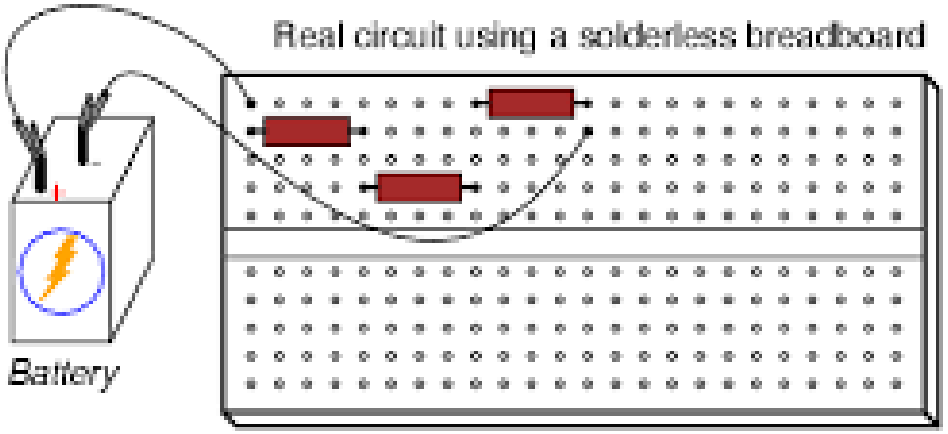
```
void setup() {
  // put your setup code here, to run once:
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(12, OUTPUT);
}

void loop() {
  static int num = 0;
  // put your main code here, to run repeatedly:
  digitalWrite(9, !!(num & (0x01<<0)));
  digitalWrite(10, !!(num & (0x01<<1)));
  digitalWrite(11, !!(num & (0x01<<2)));
  digitalWrite(12, !!(num & (0x01<<3)));
  num++;
  num = num % 10;
  delay(1000);
}
```

Breadboard





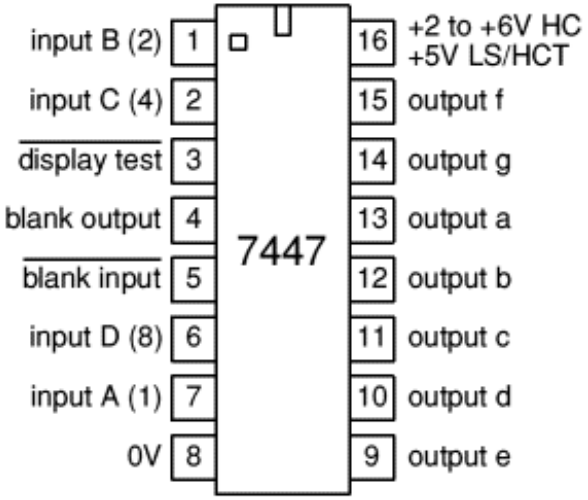




Use DC power supply on your bench to power the circuit and arduino, don't use Arduino as power source. (to protect your Arduino and your laptop)

Connect 5V to the red line and Ground to the blue line.

BCD-7 Segment Decoder



BCD--Binary Coded Decimal

- Before signals go to LED, a 330ohms resistor is needed for each output of 7447 (To limit the current, protect the LED light and your eyes)
- Every IC need to be powered. Connect Vcc (Pin 16) to 5V, and GND (Pin 8) to ground.

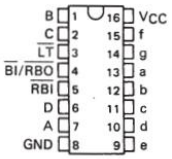
Read the datasheet

Xi'an Jiaotong-Liverpool University 西交利物浦大学

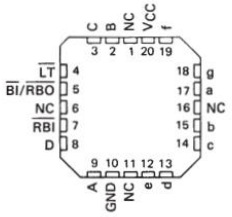
SN5446A, '47A, '48, SN54LS47, 'LS48, 'LS49
SN7446A, '47A, '48, SN74LS47, 'LS48, 'LS49
BCD-TO-SEVEN-SEGMENT DECODERS/DRIVERS
SDLS111 – MARCH 1974 – REVISED MARCH 1988

'46A, '47A, 'LS47 feature	'48, 'LS48 feature	'LS49 feature
• Open-Collector Outputs Drive Indicators Directly	• Internal Pull-Ups Eliminate Need for External Resistors	• Open-Collector Outputs
• Lamp-Test Provision	• Lamp-Test Provision	• Blanking Input
• Leading/Trailing Zero Suppression	• Leading/Trailing Zero Suppression	

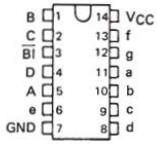
SN5446A, SN5447A, SN54LS47, SN5448, SN54LS48 . . . J PACKAGE
SN7446A, SN7447A, SN7448 . . . N PACKAGE
SN74LS47, SN74LS48 . . . D OR N PACKAGE
(TOP VIEW)



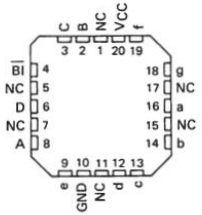
SN54LS47, SN54LS48 . . . FK PACKAGE
(TOP VIEW)



SN54LS49 . . . J OR W PACKAGE
SN74LS49 . . . D OR N PACKAGE
(TOP VIEW)



SN54LS49 . . . FK PACKAGE
(TOP VIEW)



NC — No internal connection

SN5446A, '47A, '48, SN54LS47, 'LS48, 'LS49
SN7446A, '47A, '48, SN74LS47, 'LS48, 'LS49
BCD-TO-SEVEN-SEGMENT DECODERS/DRIVERS
SDLS111 – MARCH 1974 – REVISED MARCH 1988

description

The '46A, '47A, and 'LS47 feature active-low outputs designed for driving common-anode LEDs or incandescent indicators directly. The '48, 'LS48, and 'LS49 feature active-high outputs for driving lamp buffers or common-cathode LEDs. All of the circuits except 'LS49 have full ripple-blanking input/output controls and a lamp test input. The 'LS49 circuit incorporates a direct blanking input. Segment identification and resultant displays are shown below. Display patterns for BCD input counts above 9 are unique symbols to authenticate input conditions.

The '46A, '47A, '48, 'LS47, and 'LS48 circuits incorporate automatic leading and/or trailing-edge zero-blanking control ($\overline{RBI}$ and $\overline{RBO}$). Lamp test ($\overline{LT}$) of these types may be performed at any time when the $\overline{BI}/\overline{RBO}$ node is at a high level. All types (including the '49 and 'LS49) contain an overriding blanking input ($\overline{BI}$), which can be used to control the lamp intensity by pulsing or to inhibit the outputs. Inputs and outputs are entirely compatible for use with TTL logic outputs.

The SN54246/SN74246 and '247 and the SN54LS247/SN74LS247 and 'LS248 compose the 6 and the 9 with tails and were designed to offer the designer a choice between two indicator fonts.

SEGMENT IDENTIFICATION

NUMERICAL DESIGNATIONS AND RESULTANT DISPLAYS

'46A, '47A, 'LS47 FUNCTION TABLE (T1)

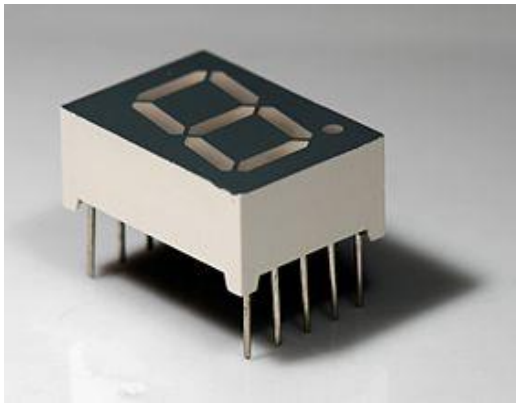
DECIMAL OR FUNCTION	INPUTS $\overline{LT}$	$\overline{RBI}$	D	C	B	A	$\overline{BI}/\overline{RBO}$†	OUTPUTS a	b	c	d	e	f	g	NOTE
0	H	H	L	L	L	L	H	ON	ON	ON	ON	ON	ON	OFF	
1	H	X	L	L	L	H	H	OFF	ON	ON	OFF	OFF	OFF	OFF	
2	H	X	L	L	H	L	H	ON	ON	OFF	ON	ON	OFF	ON	
3	H	X	L	L	H	H	H	ON	ON	ON	ON	OFF	OFF	ON	
4	H	X	L	H	L	L	H	OFF	ON	ON	OFF	OFF	ON	ON	
5	H	X	L	H	L	H	H	ON	OFF	ON	ON	OFF	ON	ON	
6	H	X	L	H	H	L	H	OFF	OFF	ON	ON	ON	ON	ON	
7	H	X	L	H	H	H	H	ON	ON	ON	OFF	OFF	OFF	OFF	1
8	H	X	H	L	L	L	H	ON	ON	ON	ON	ON	ON	ON	
9	H	X	H	L	L	H	H	ON	ON	ON	OFF	OFF	ON	ON	
10	H	X	H	L	H	L	H	OFF	OFF	OFF	ON	ON	OFF	ON	
11	H	X	H	L	H	H	H	OFF	OFF	ON	ON	OFF	OFF	ON	
12	H	X	H	H	L	L	H	OFF	ON	OFF	OFF	OFF	ON	ON	
13	H	X	H	H	L	H	H	ON	OFF	OFF	ON	OFF	ON	ON	
14	H	X	H	H	H	L	H	OFF	OFF	OFF	ON	ON	ON	ON	
15	H	X	H	H	H	H	H	OFF	OFF	OFF	OFF	OFF	OFF	OFF	
BI	X	X	X	X	X	X	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	2
RBI	H	L	L	L	L	L	L	OFF	OFF	OFF	OFF	OFF	OFF	OFF	3
LT	L	X	X	X	X	X	H	ON	ON	ON	ON	ON	ON	ON	4

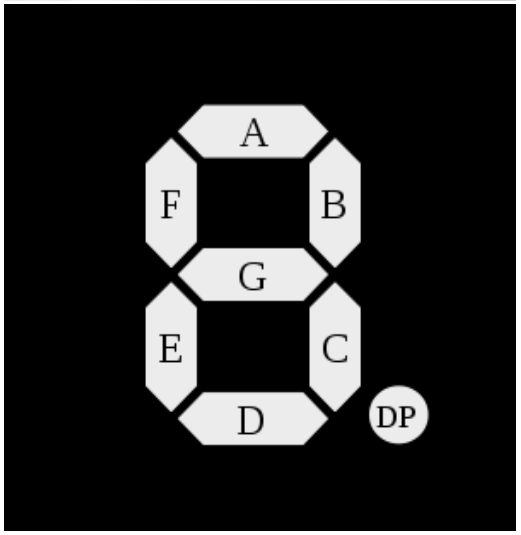
H = high level, L = low level, X = irrelevant

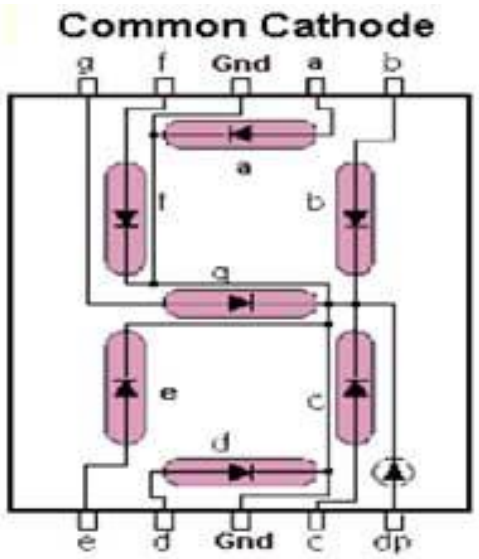
NOTES: 1. The blanking input ($\overline{BI}$) must be open or held at a high logic level when output functions 0 through 15 are desired. The ripple-blanking input ($\overline{RBI}$) must be open or high if a 0 in the display is not desired.
2. When a low logic level is applied directly to the blanking input ($\overline{BI}$), all segment outputs are off regardless of the level of any other input.
3. When ripple-blanking input ($\overline{RBI}$) and inputs A, B, C, and D are at a low level with the lamp test input high, all segment outputs go off and the ripple-blanking output ($\overline{RBO}$) goes to a low level (response condition).
4. When the blanking input/ripple blanking output ($\overline{BI}/\overline{RBO}$) is open or held high and a low is applied to the lamp-test input, all segment outputs are on.

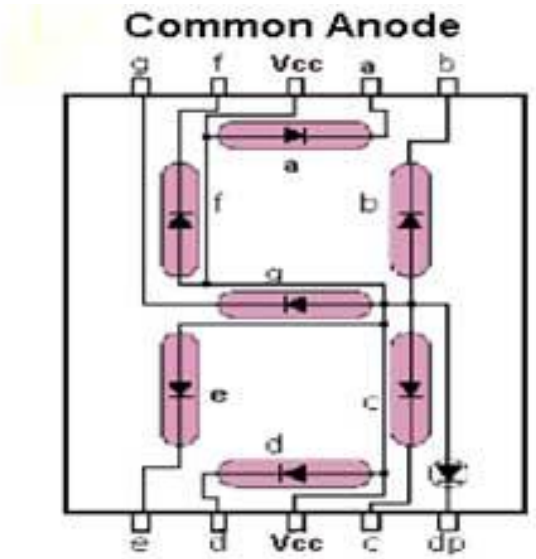
†$\overline{BI}/\overline{RBO}$ is wire AND logic serving as blanking input ($\overline{BI}$) and/or ripple-blanking output ($\overline{RBO}$).

7 Segment Indicator







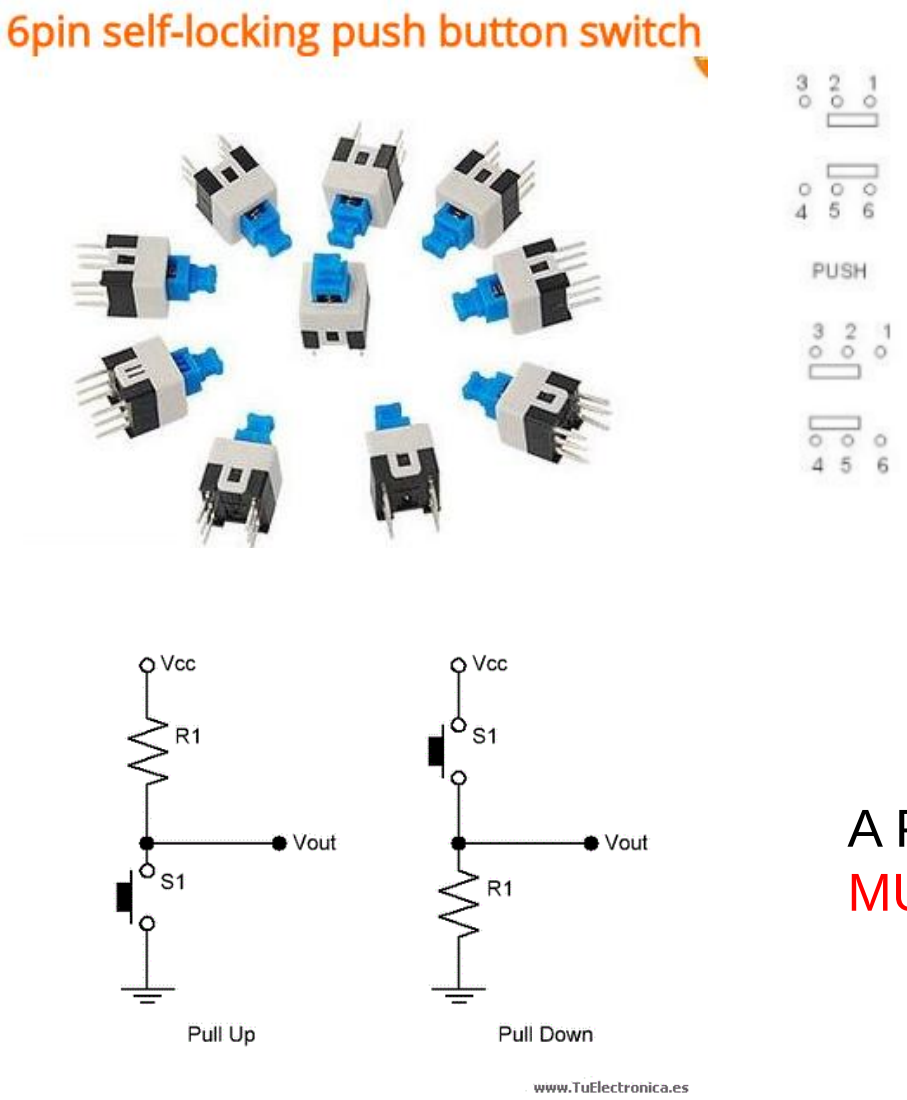


How to light up number '7' in Common Cathode LED.

a	b	c	d	e	f	g
1	1	1	0	0	0	0

How to light up number '4' in Common Anode LED ?

Using switch



digitalRead_key | Arduino 1.8.1

File Edit Sketch Tools Help

digitalRead_key §

```
void setup() {
Serial.begin(9600);
pinMode(7,INPUT); //Stop key
pinMode(6,INPUT); //Reset key
}
void loop() {
int StopKey = digitalRead(7);
int ResetKey = digitalRead(6);

Serial.print("Stop Key"); //test Only
Serial.println(StopKey); //click the serial monitor to see
                         //the value of this variable
Serial.print("Reset Key"); //test Only
Serial.println(ResetKey); //test Only
}
```

Done uploading.

A Pull up or Pull down Resistor MUST (around 10K Ω) be used.

Basic requirements

Xi'an Jiaotong-Liverpool University 西交利物浦大學

1. Two digit displays:

   Your clock has to display two digit numbers. Each digit should roll over at a certain time.

   The roll over time is your group number, if you are group number is single digit, then your roll over time is 50 + group number.

   For example,

   - Your are in group B34, roll over time is 34. Your clock should count 33, 34, 00, 01, 02 …
   - Your are in group A05, roll over time is 50+05=55. Your clock should count 54, 55, 00, 01, 02 …

2. Pause and reset:

   Your clock must can be stopped and reset by press bottoms.

Files to Be Uploaded

Xi'an Jiaotong-Liverpool University 西交利物浦大學

On behalf of the whole team, the team leader should upload one video clip to ICE.

MAXIMUM SIZE FOR UPLOADING: 20MB

Schudule

Xi'an Jiaotong-Liverpool University 西交利物浦大學

Week	Lab name	Student group	Assessment
Week-8	Digital clock	**Group A**	Bench inspection for Digital clock (**Group A**)
	Matlab computer lab 2	**Group B**	
Week-9	Digital clock	**Group B**	Bench inspection for Digital clock (**Group B**)
	Matlab computer lab 2	**Group A**	

Assessment (5%)

Xi'an Jiaotong-Liverpool University 西交利物浦大學

Group marks - Bench inspection			Total
Two-digit roll over (2%)	Stop and reset (2%)	Circuit layout (1%)	5%

Bench inspection: demonstration of stopwatch function and circuit layout.