

Embedded systems, microcontrollers, programming and applications



Embedded Systems

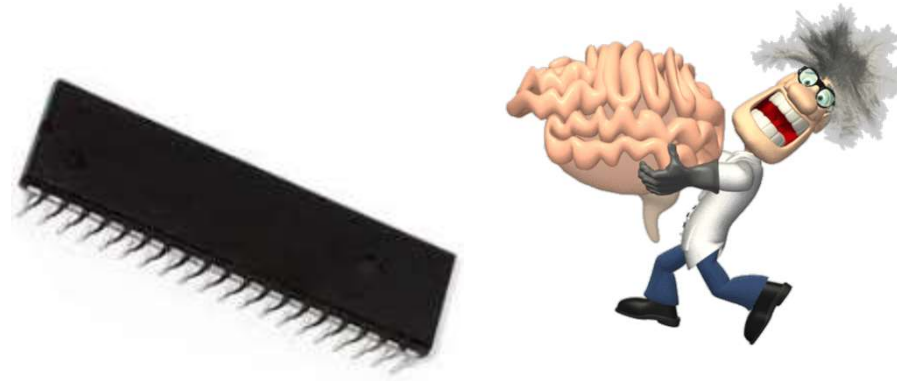
- = a computer system that performs one/a few **dedicated functions**.
- Must **respond in time** to external stimulus.
- Often includes hardware / **mechanical parts**.
- Is controlled by a **micro-controller / DSP**.
- Design engineers **optimize** it (size, cost, reliability, performance).
- E.g. MP3 player, traffic light controller,



More embedded systems

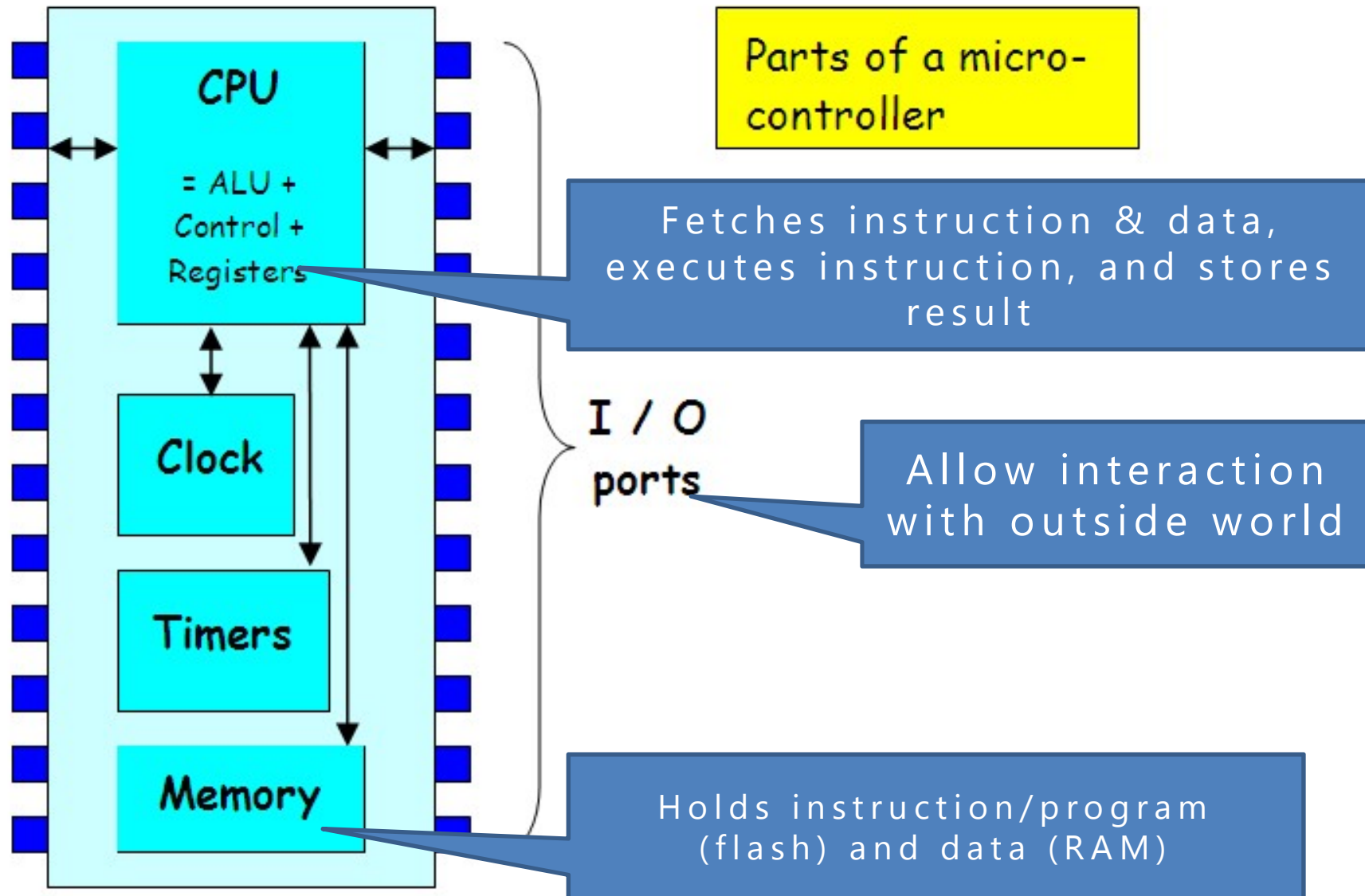


What is a microcontroller?

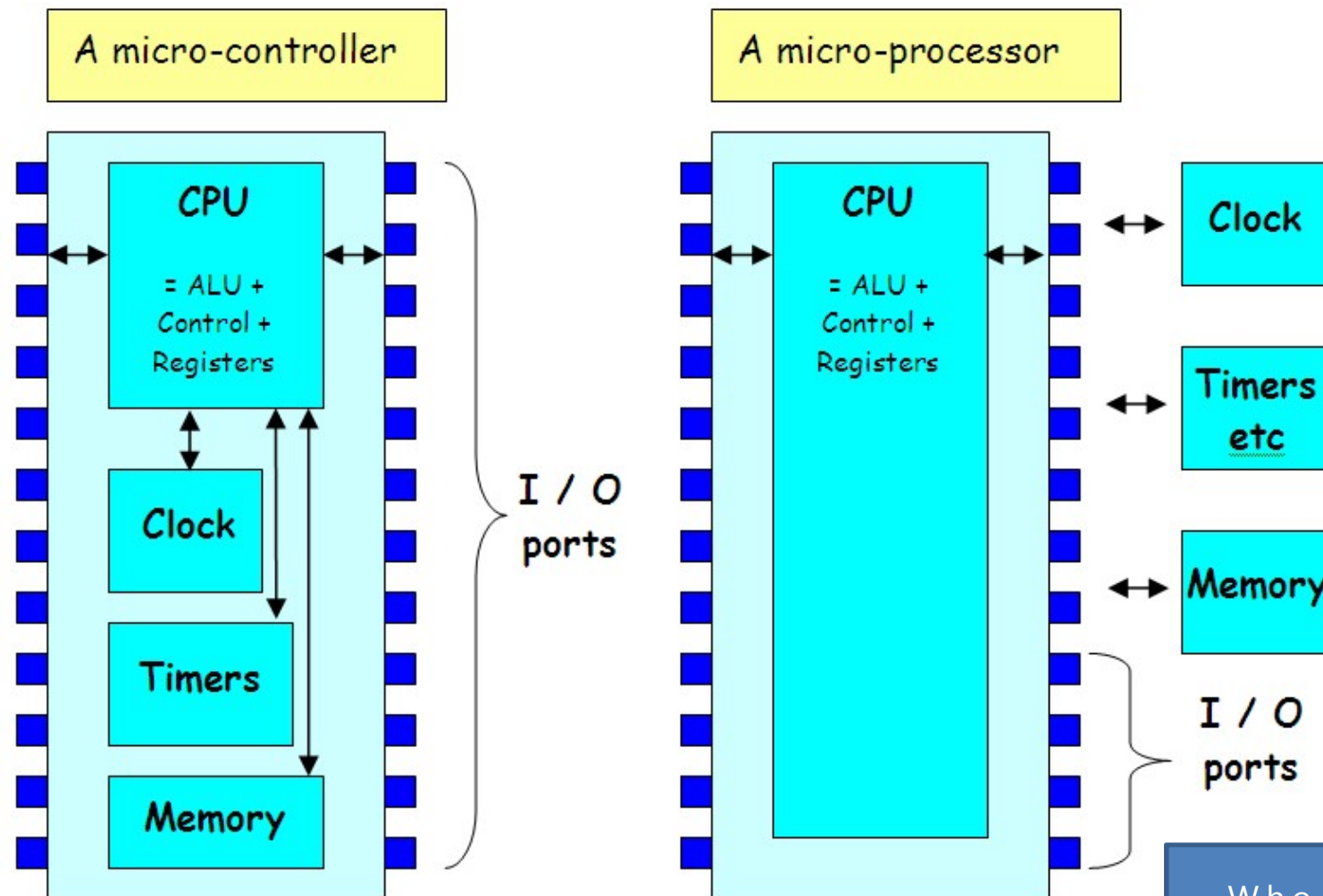


- A small computer on a single integrated circuit
- = CPU + clock + timers + IO ports + memory
- For dedicated applications

Main parts of a micro-controller



How is a micro-controller different from a micro-processor?



Who is cheaper?
Who is smaller?

Monitor

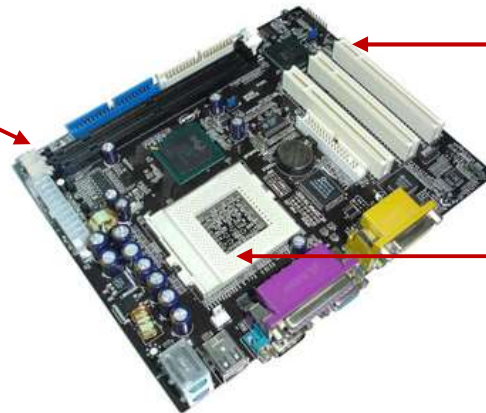


CPU
(Central
Processing
Unit)

Mouse

Keyboard

Peripheral
devices



Connections
for other I/O
peripherals

**Micro-
processor**

Mother board

Your PC...

Common characteristics of a micro-controller

Cheap

Small

Low
power

E.g. the micro-controller
EMBEDDED inside an MP3
player

Dedicated

Fixed IO



The development tools & programming languages for a micro-controller

Hardware development

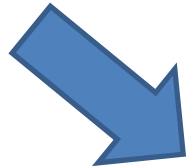
Select IO devices (& uC)

- Design circuits
- Fabricate circuits
- Interface circuits

Software development

Used IDE to create a project

- Write a C-program
- Compile it into machine code
- Download into the uC



Integration / testing / debugging

Programming

Example high level program :

```
if (temp > 30)
    on_motor;
```

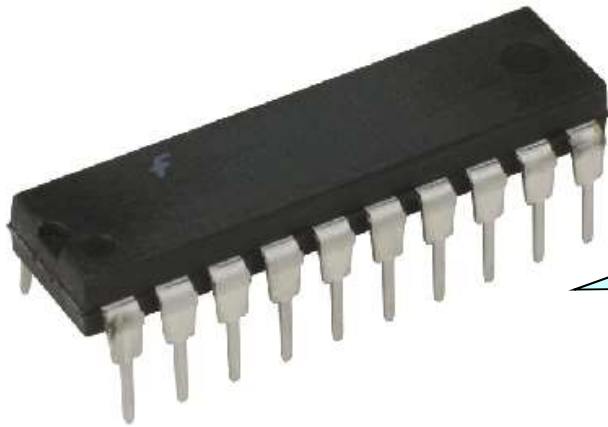
Example assembly code :

```
BSF          03h, 5          ; go to Bank 1
MOVLW 06h     ; put 00110 into W
MOVWF 85h     ; move 00110 into TRISA
BCF          03h, 5          ; come back to
Bank 0
```

Microcontroller understands “machine code”

```
// C - program  
if (temp > 30)  
    on_motor;
```

“Compile”



101010111110101
0000100101110...
....

Hex file

Why Interface?

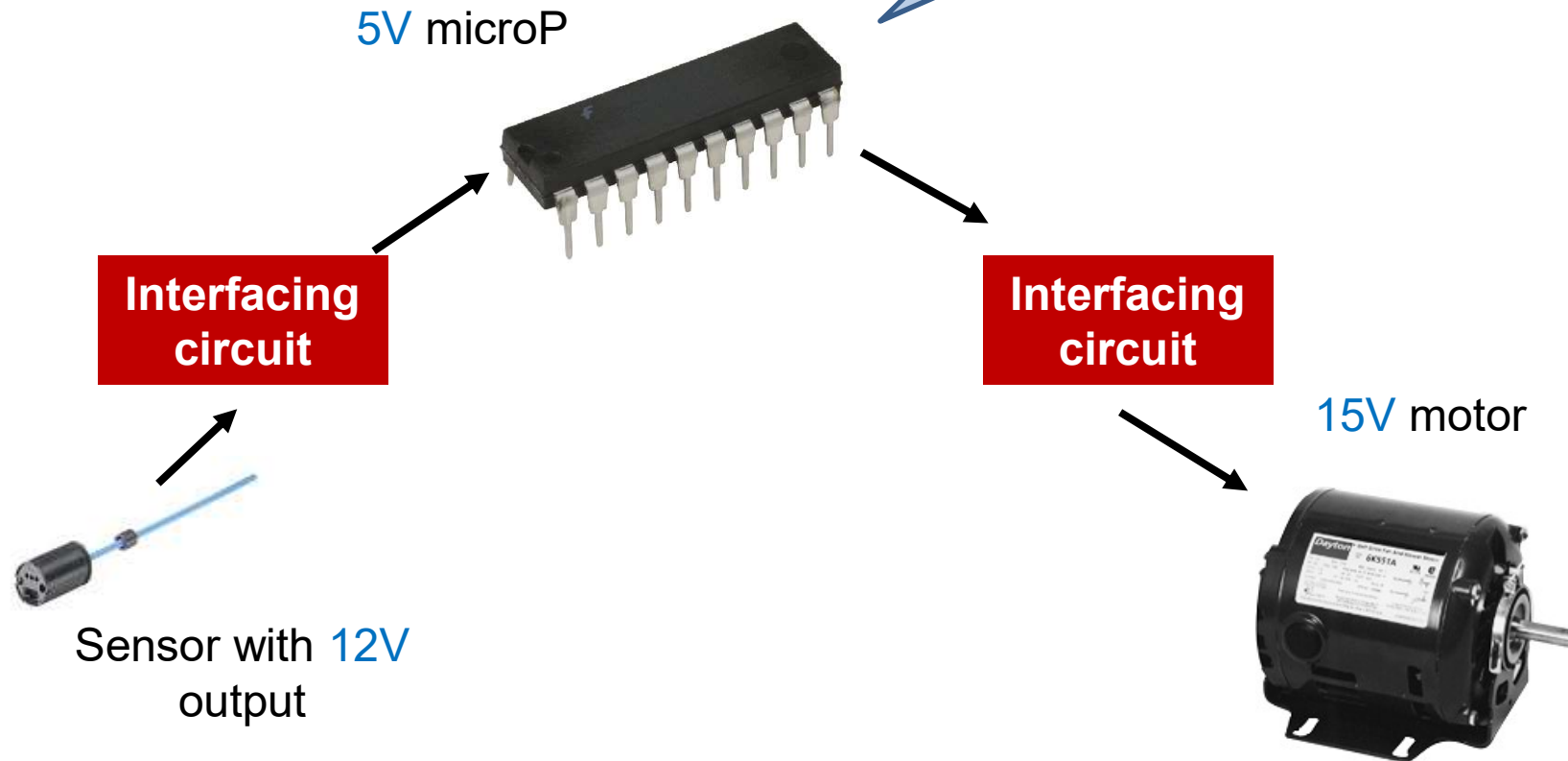
- Voltage / current mismatch
- Limited no of I/O pins



Interfacing reasons

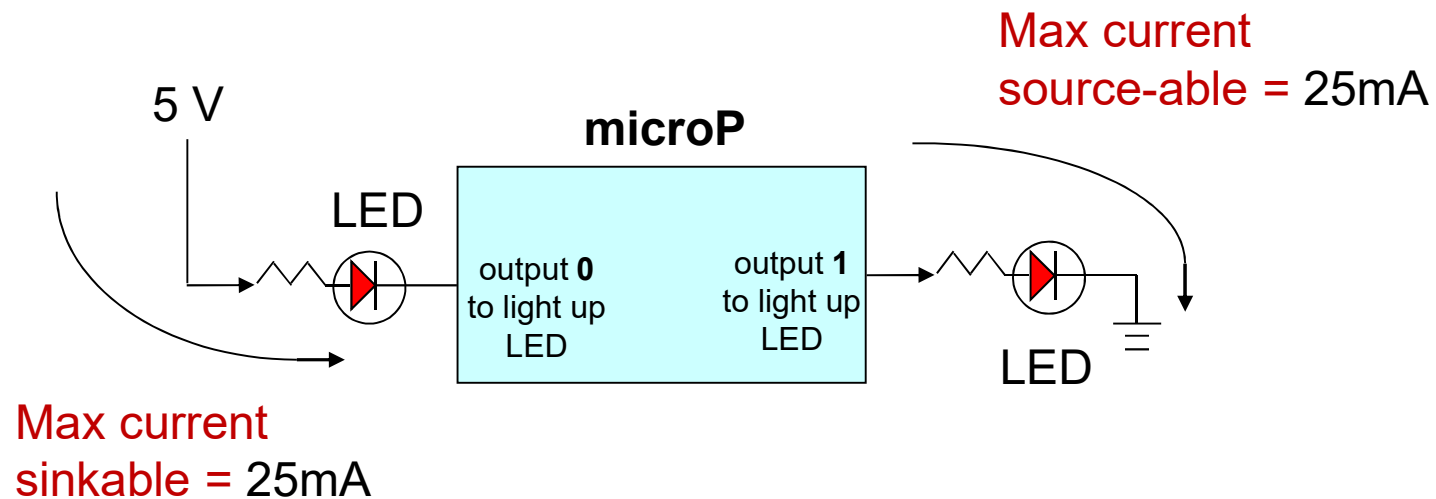
Voltage / current m

Micro controller 3.3V or 5V device
So, need additional circuitry to
connect to higher voltage I/O devices



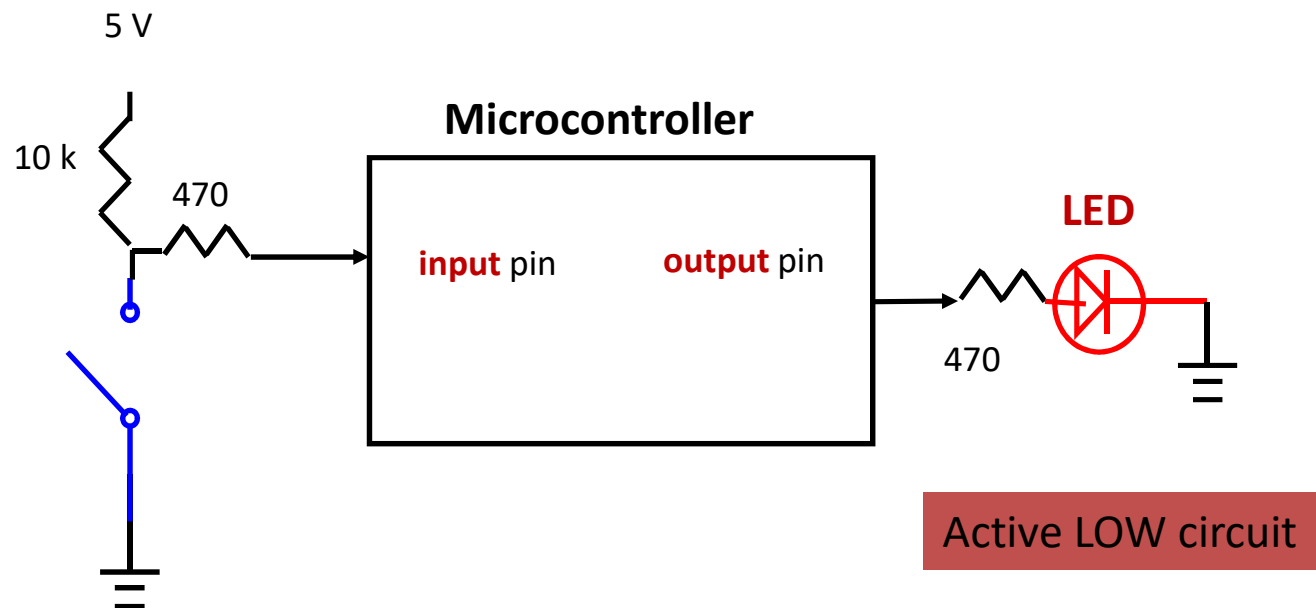
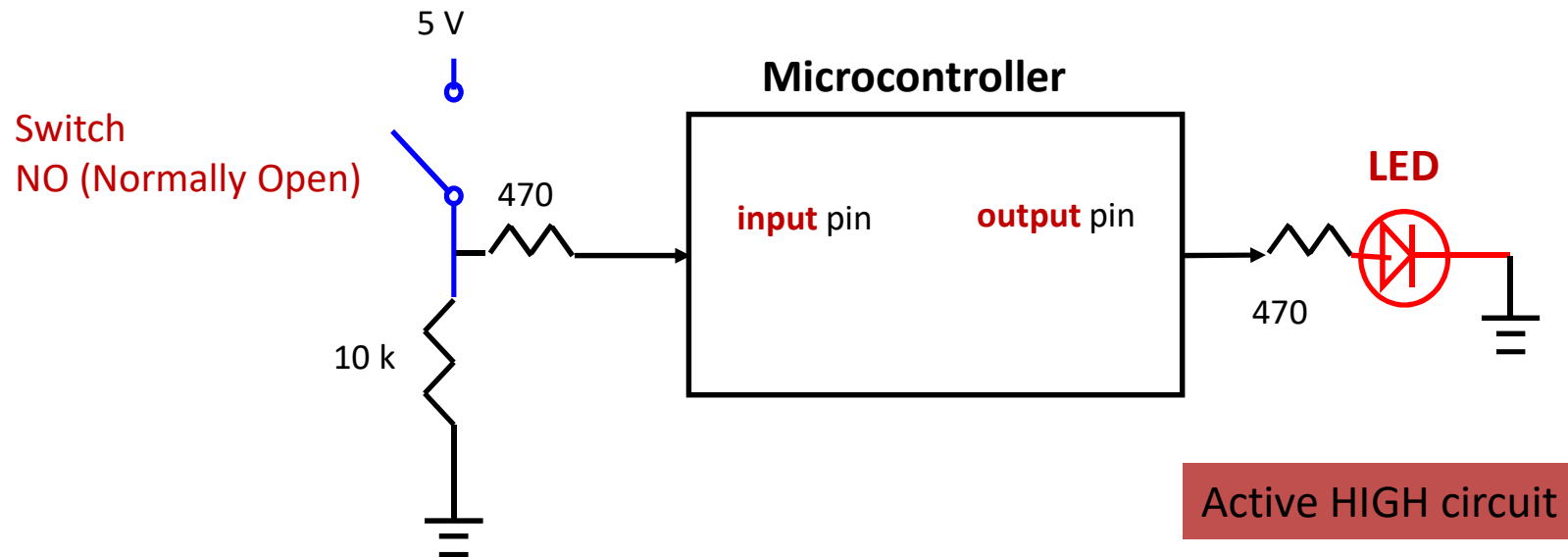
Interfacing reasons

Voltage / current mismatch

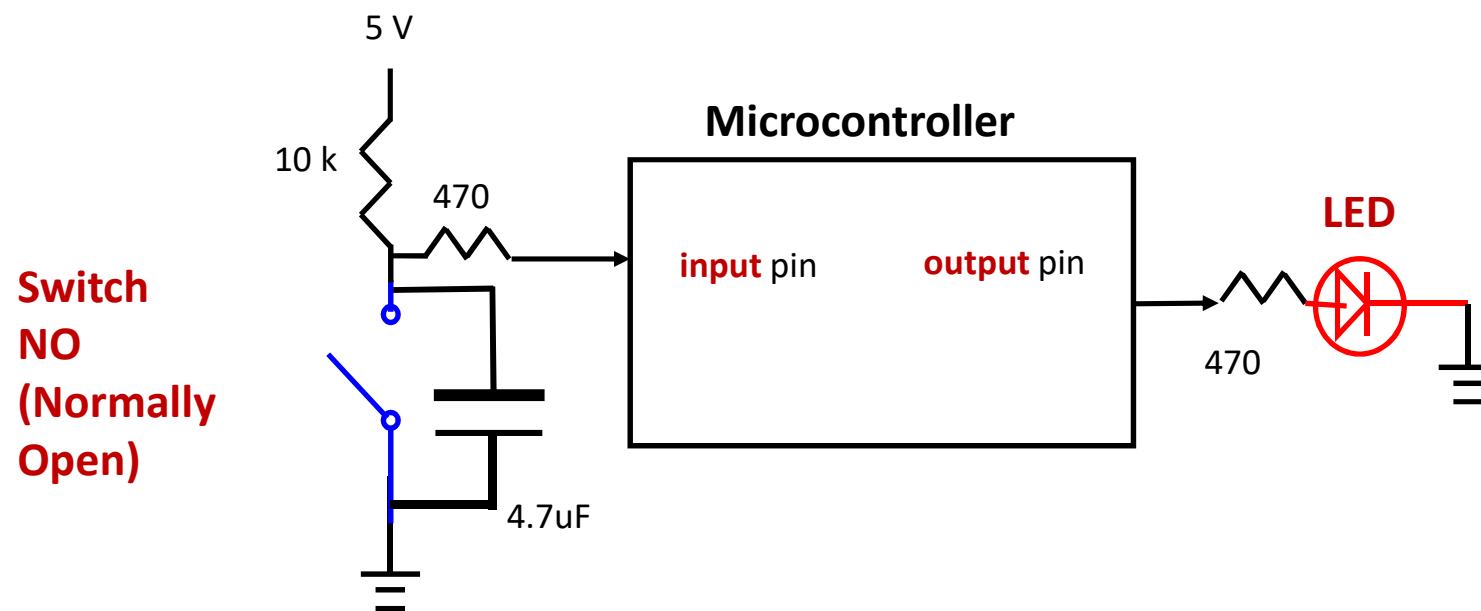


Each pin can sink / source **25 mA**.
All the pins together can sink / source **200 mA**.

Connecting a switch to digital I/O



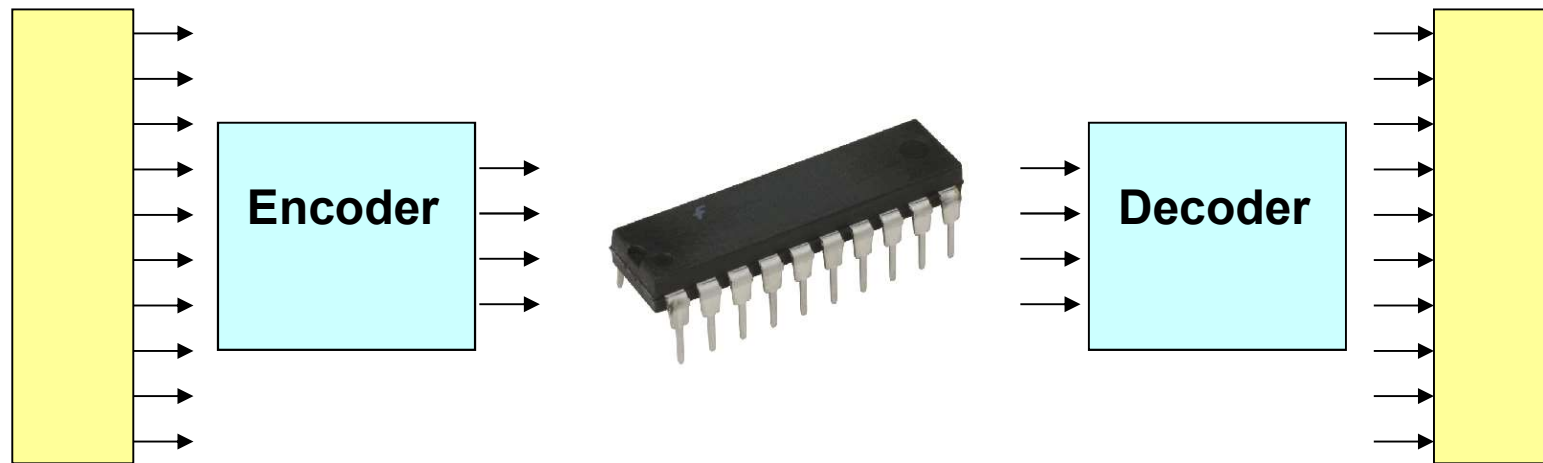
Connecting a switch to digital I/O



Active LOW with noise filter circuit

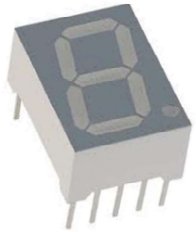
Interfacing reasons

Limited no of I/O pins



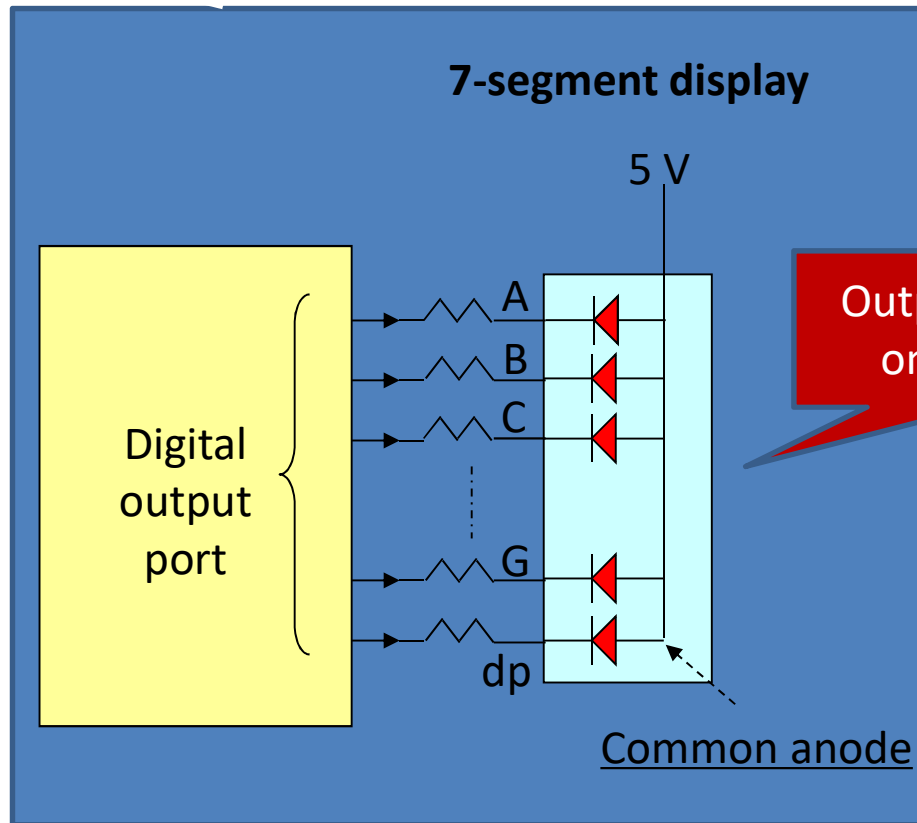
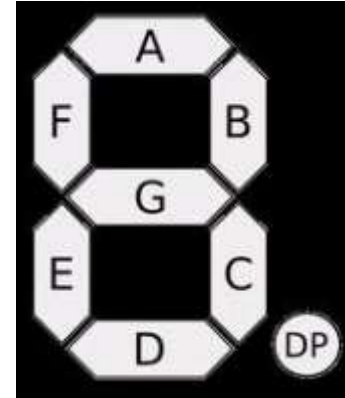
Encoder / decoder reduces number of I/O pins required to interface to a device.

7-segment display



How it looks

How the segments are labelled



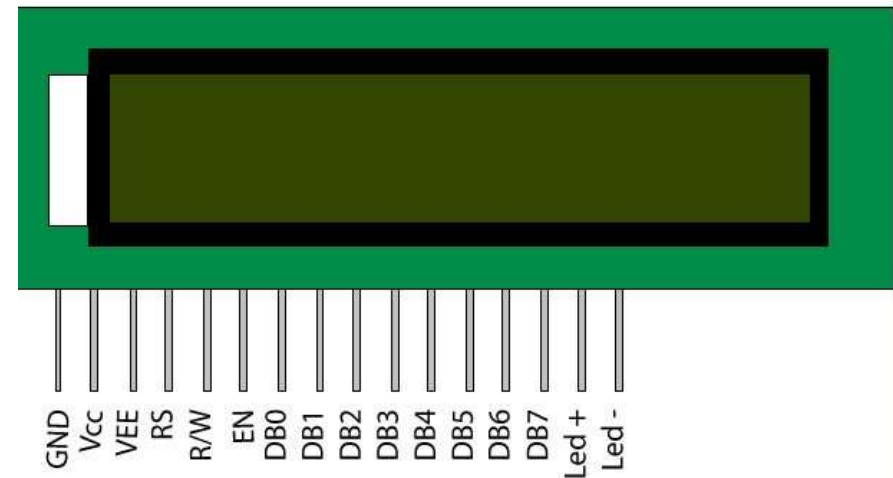
Application: Q-number display



LCD

Pin Description:

Name	Function
Ground	Ground (0V)
V _{CC}	Supply voltage; 5V (4.7V – 5.3V)
V _{EE}	Contrast adjustment; through a variable resistor
Register Select	Selects command register when low; and data register when high
Read/write	Low to write to the register; High to read from the register
Enable	Sends data to data pins when a high to low pulse is given
DB0	8-bit data pins
DB1	
DB2	
DB3	
DB4	
DB5	
DB6	
DB7	
Led+	Backlight V _{CC} (5V)
Led-	Backlight Ground (0V)

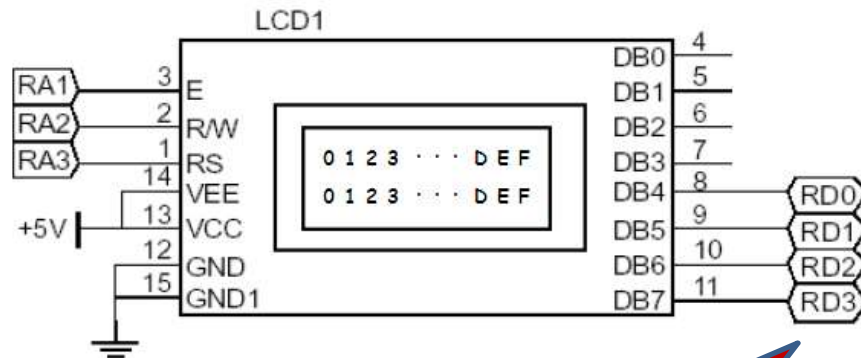


Application:
display 'prompt'



How it looks

LCD




4-bit mode: 1 byte written as 2 nibbles (each 4 bits), via RD3-0



In 8-bit mode, bytes can be written at a faster rate

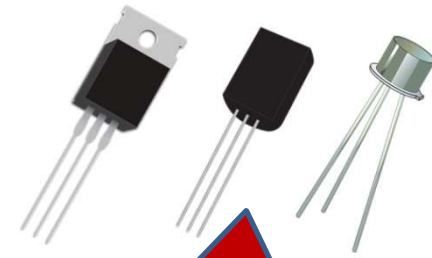
In 8-bit mode, more PIC pins needed for connection



What are the pros & cons of the **8-bit** mode? 



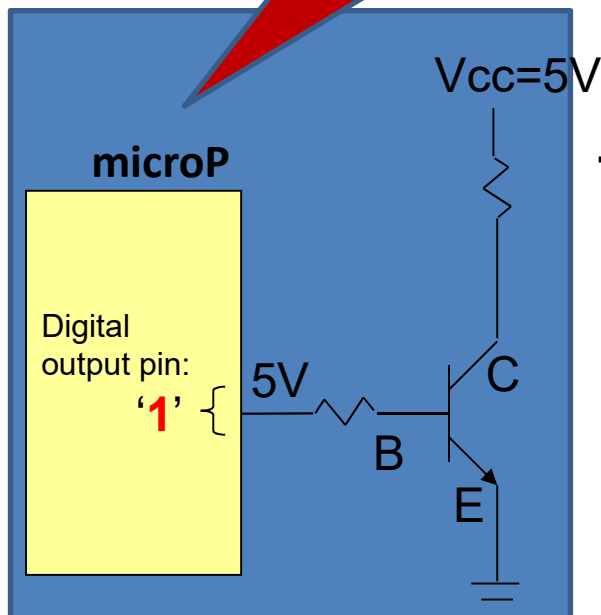
Transistor as a switch



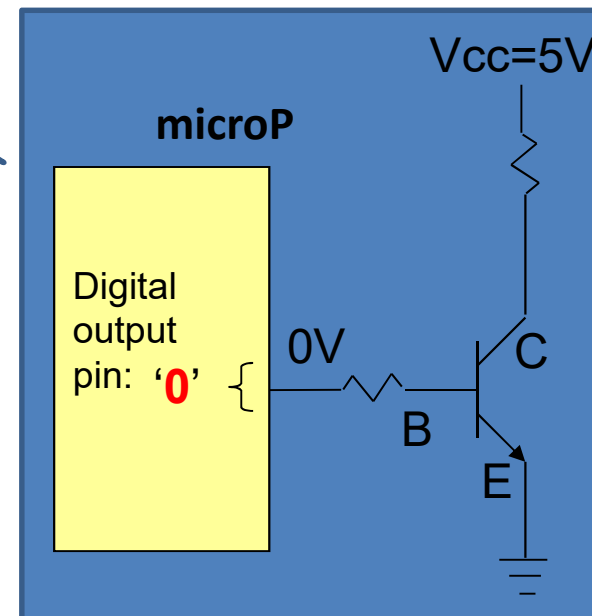
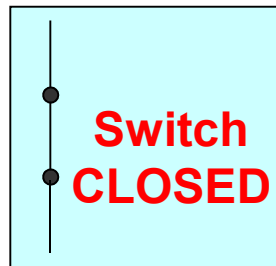
How it looks

Output '0' to turn off a transistor

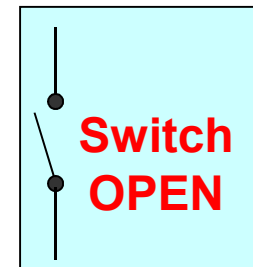
Output '1' to turn on a transistor



Transistor ON



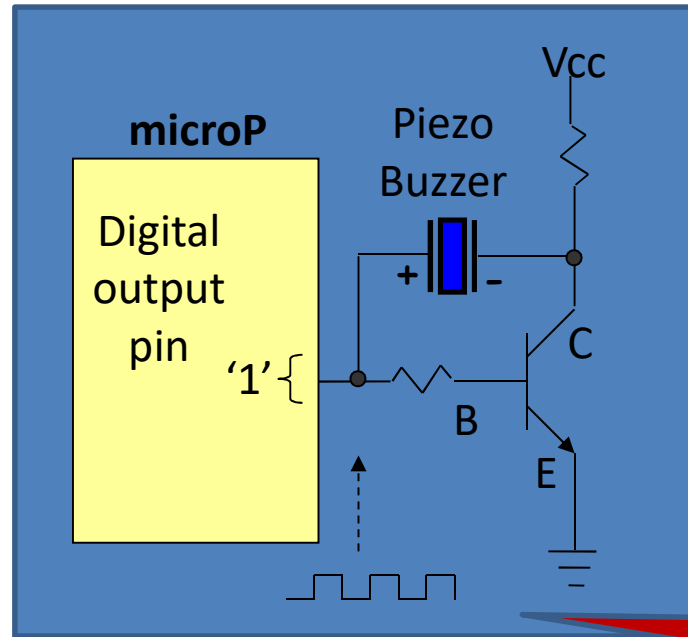
Transistor OFF



Applications: to turn on or off a buzzer, a motor, or a solenoid

Buzzer

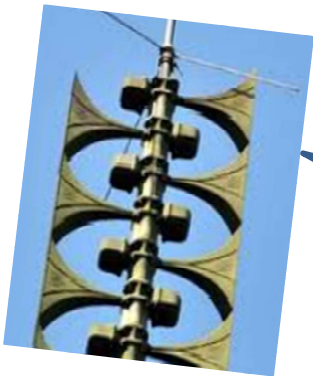
How it looks



"Pulse train" turns buzzer on & off → buzzing tone.

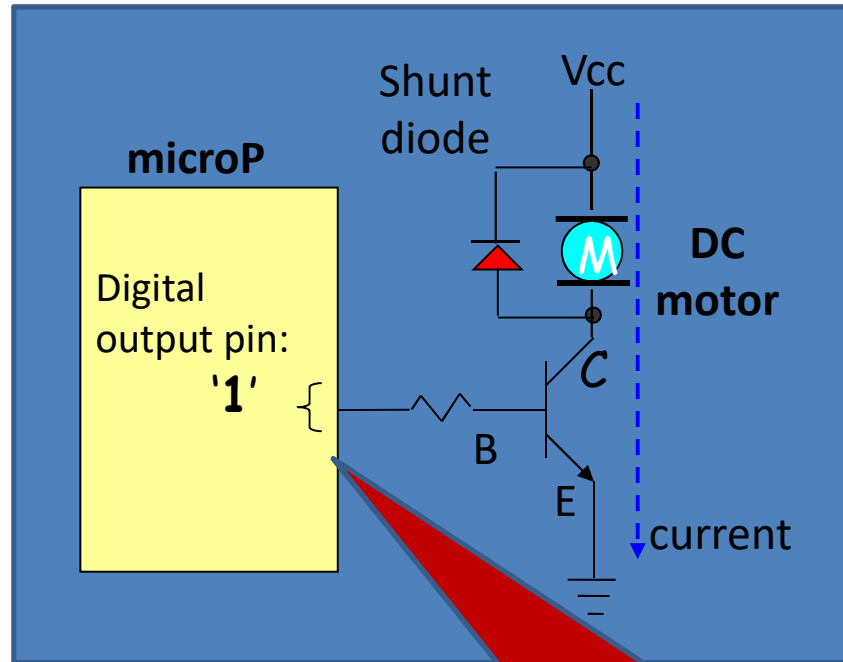
Can vary the pitch...

Application: alert, warning...



<https://www.youtube.com/watch?v=kqv-aan4pBU>

DC motor ON/OFF



How it looks



Application: when 'movement' is required

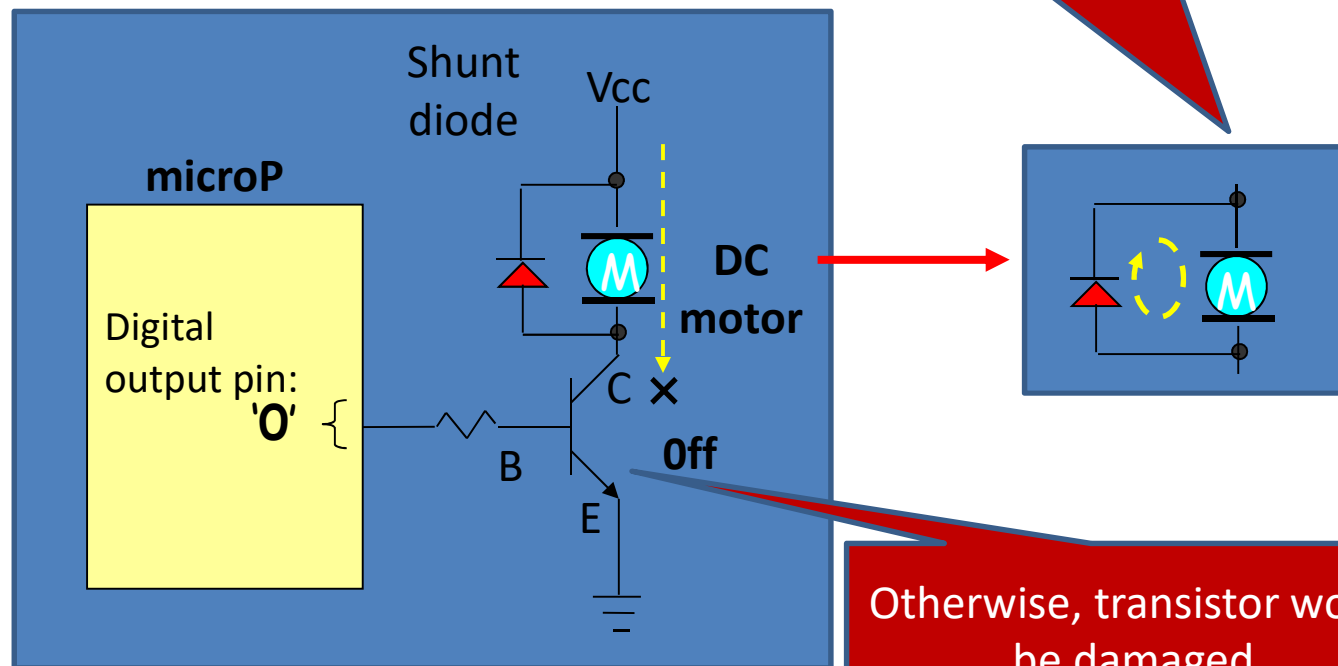
Output '1' to turn on a motor.



DC motor ON/OFF

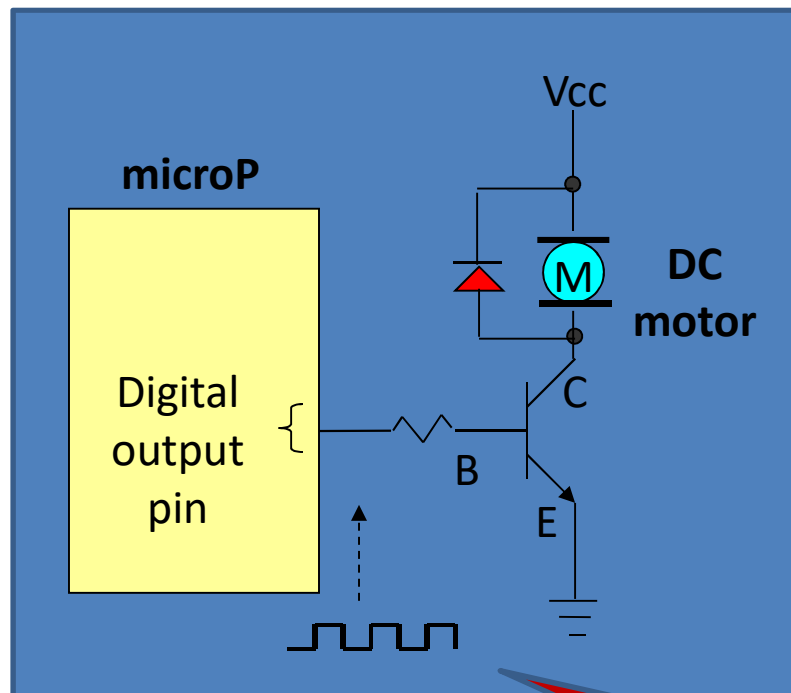
Why Shunt diode??

When transistor is turned from ON to OFF, **shunt diode** allows current to continue to flow for a short while.



Otherwise, transistor would be damaged.

PWM (Pulse Width Modulation) for speed control

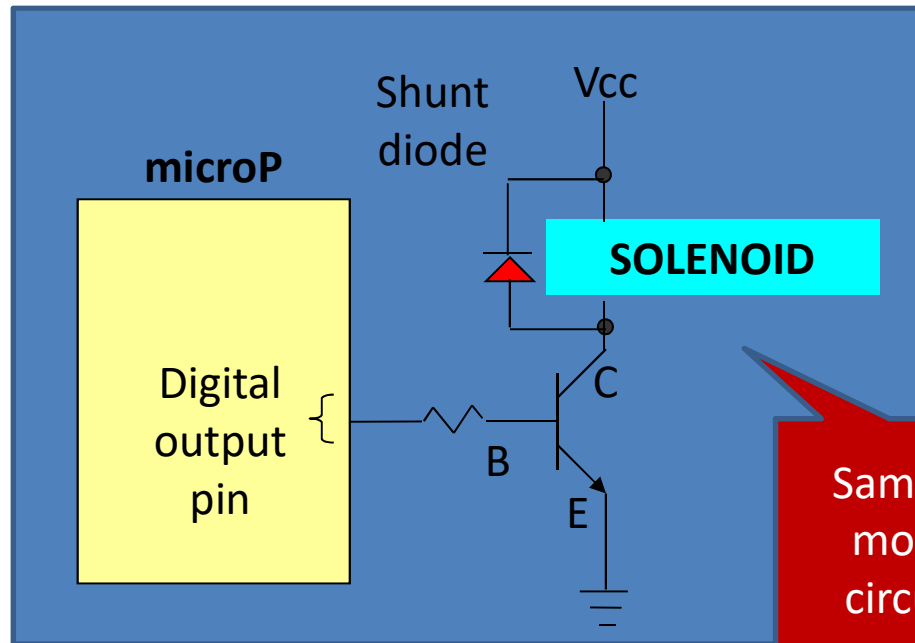


Duty cycle = high time / period

By varying the duty cycle, the motor speed can be controlled.

Duty cycle = 50%

Motor circuit for solenoid



How it looks



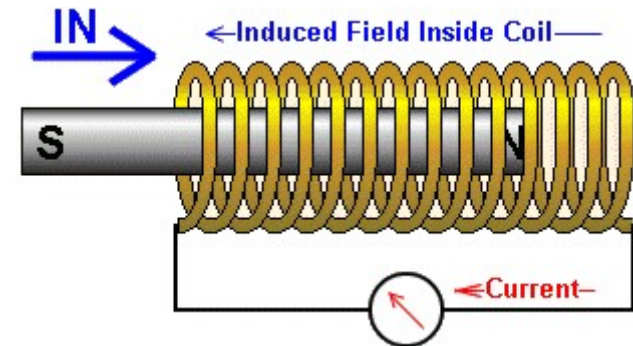
Solenoid = **electromagnet**

When energised, suck in (or push out) iron bar.

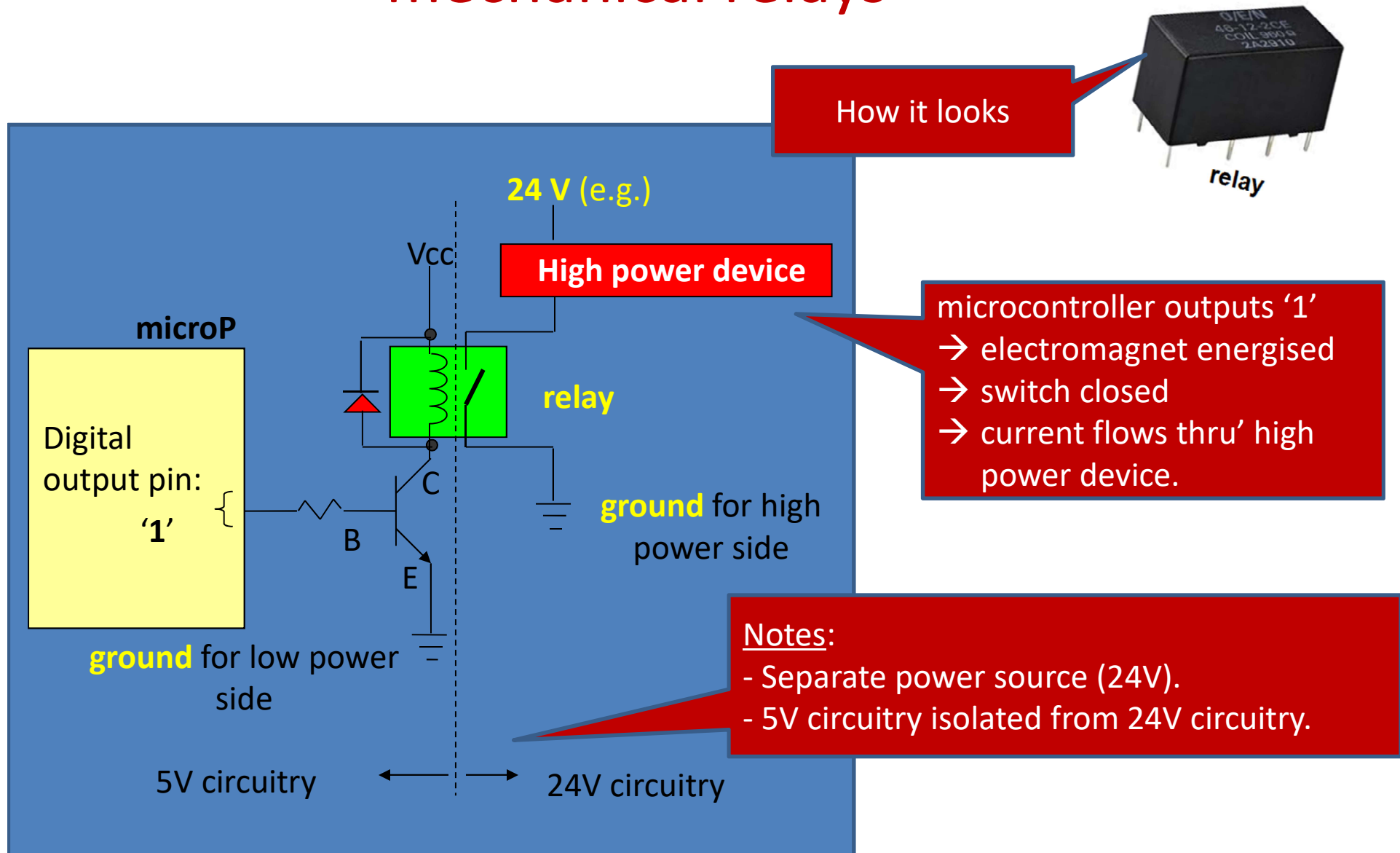
Same as motor circuit!



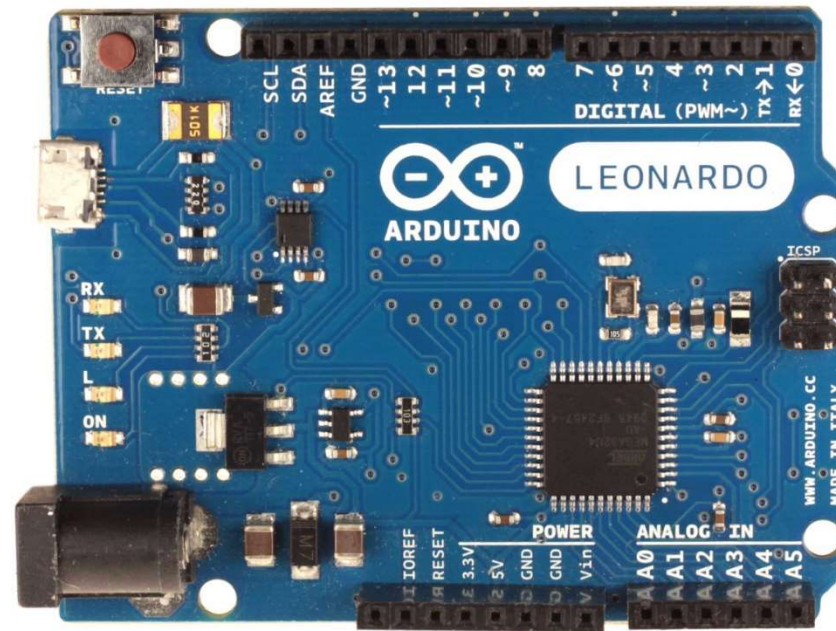
Application: electronic lock, or solenoid valve (to control liquid flow)



Driving high power devices – via mechanical relays



What is Arduino?

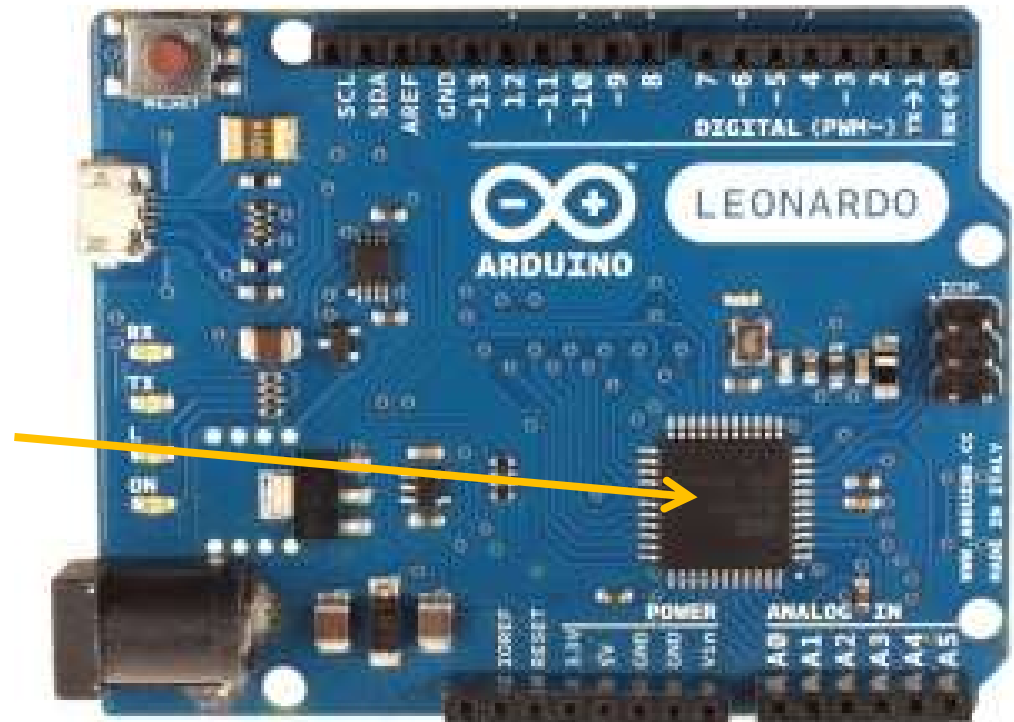


- Microcontroller ATmega328
- Operating Voltage 5V
- Input Voltage (recommended) 7-12V
- Input Voltage (limits) 6-20V
- Digital I/O Pins 14 (of which 6 provide PWM output)
- Analog Input Pins 6
- DC Current per I/O Pin 40 mA
- DC Current for 3.3V Pin 50 mA
- Flash Memory 32 KB (ATmega328) of which 0.5 KB used by bootloader
- SRAM 2 KB (ATmega328)
- EEPROM 1 KB (ATmega328)
- Clock Speed 16 MHz
- Length 68.6 mm
- Width 53.4 mm
- Weight 25 g



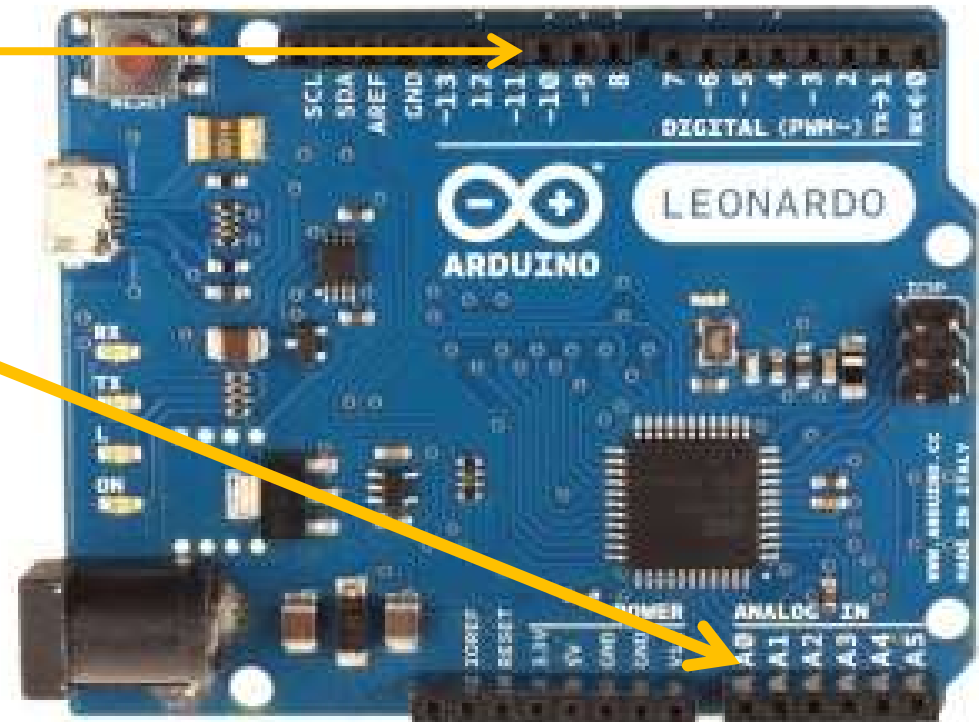
Arduino memory

- Flash memory (program space), is where the Arduino sketch is stored.
- SRAM (static random access memory) is where the sketch creates and manipulates variables when it runs.
- EEPROM is memory space that programmers can use to store long-term information.



Arduino I/O

Digital I/O Pins 14
Analog Input Pins 6



ACTIVITY- Programming and learning Arduino

You will need:

- Laptop
- Arduino
- breadboard
- LED
- 200ohm resistor
- Push button
- 4.7kohm resistor
- 1uF capacitor
- Oscilloscope (optional)

OR simply simulate using tinkercad.com

Try all the example programs



mfercan

Search designs...

3D Designs

Circuits

Codeblocks

NEW

Lessons

Your Classes

Projects

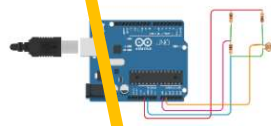


[Tinkercad Lesson Plans](#)

Tinkercad lesson plans are ready to use online or in the classroom. Discover curriculum developed in partnership with teachers. [Learn more](#)

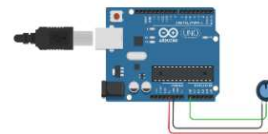
Circuits

Create new Circuit



Editing Components

7 days ago
Private



Tremendous Fulffy-Kasi

6 months ago
Private



Stunning Hillar-Kieran

7 months ago
Private



Create an account as student. Its free

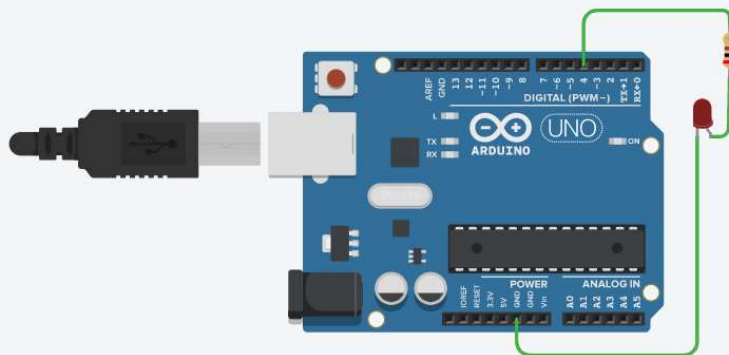
Create new circuit design with Arduino



Code

Start Simulation

Export



Text




1 (Arduino)

```
1 ///////////////////////////////////////////////////
2 //
3 //   DEMO PROGRAM
4 //
5 //
6 ///?////////////////////////////////////
7
8 int led_red = 4; // the red LED is connected to Pin 4
9
10
11 void setup() {
12   pinMode(led_red, OUTPUT);
13 }
14
15
16
17 void loop() {
18   digitalWrite(led_red, HIGH);
19   delay (1000);
20   digitalWrite(led_red, LOW);
21   delay (1000);
22 }
```

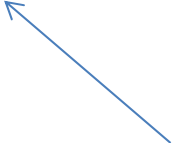
1.Lets start with basics

You need to
configure
Arduino first



```
void setup() {  
  // initialize digital pin 5 as an output.  
  pinMode(5, OUTPUT);  
}
```


```
void loop() {  
  digitalWrite(5, HIGH);    // turn the LED on  
  delay(1000);              // wait for a second  
  digitalWrite(5, LOW);    // turn the LED off  
  delay(1000);              // wait for a second  
}
```



Code here
repeats
indefinitely

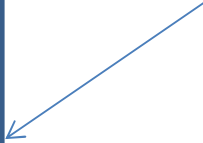
2.Can I make my loop function?

You need to
configure
Arduino first



```
void setup() {  
  // initialize digital pin 4 as an output.  
  pinMode(5, OUTPUT);  
  run_my_code();  
}
```

My code here
repeats
indefinitely



```
void run_my_code() {  
  int i=0;  
  do {  
    digitalWrite(5, HIGH);    // turn the LED on  
    delay(300);               // wait  
    digitalWrite(5, LOW);     // turn the LED off  
    delay(300);               // wait  
  }while (i==0);  
}
```

```
void loop() {}
```



I don't use
Arduino loop

3. Sometimes we don't need infinite loops?

```
void setup() {  
  // initialize digital pin 4 as an output.  
  pinMode(5, OUTPUT);  
  run_my_code();  
}
```

You need to
configure
Arduino first

```
void run_my_code() {  
  int i;  
  for (i=0;i<10;i++){  
    digitalWrite(5, HIGH);    // turn the LED on  
    delay(300);               // wait for a while  
    digitalWrite(5, LOW);     // turn the LED off  
    delay(300);               // wait for a while  
  }  
}
```

Sometimes
we don't
need infinite
loops.

LED blink
repeats only
10 times

```
void loop() {}
```

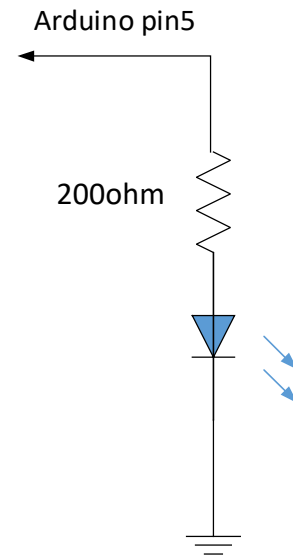
I don't use
Arduino loop

4. What about Analog output?

You need to
configure
Arduino first

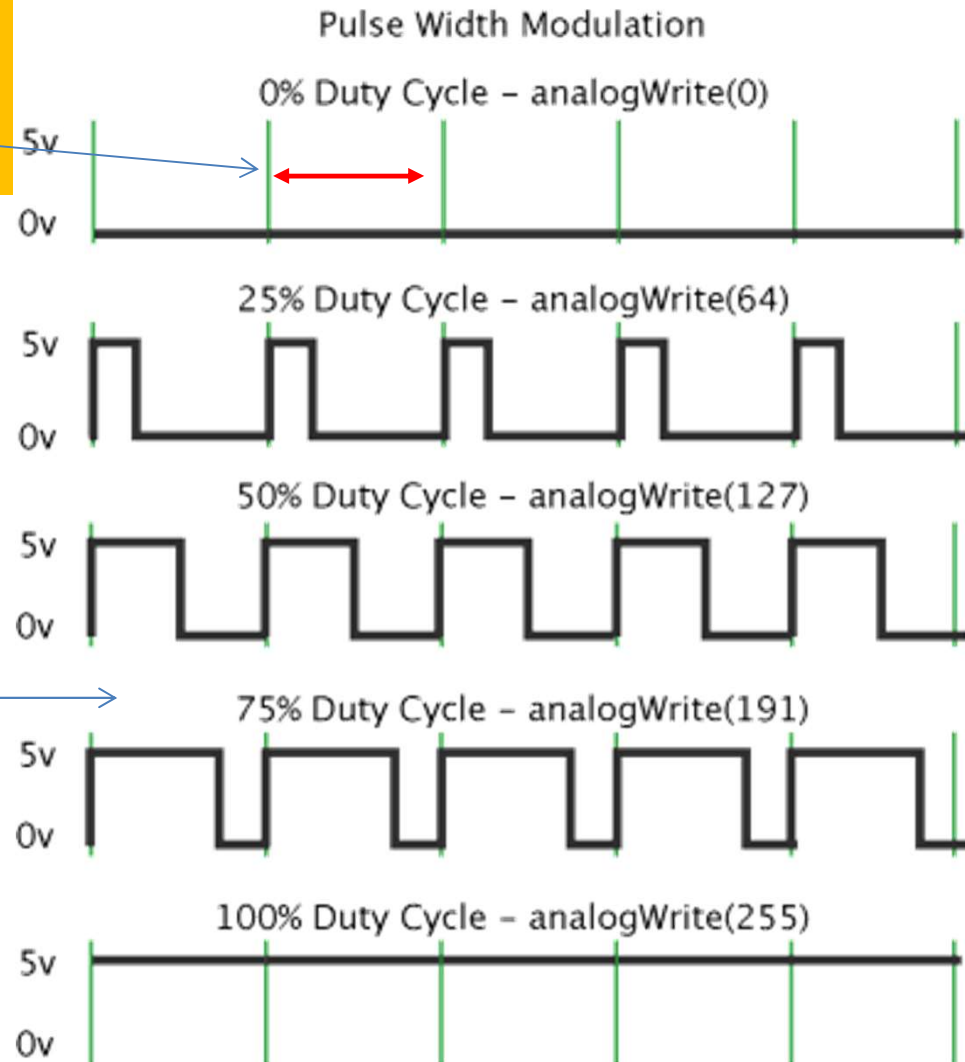
```
void setup() {  
  // initialize digital pin 4 as an output.  
  pinMode(5, OUTPUT);  
}
```

```
void loop() {  
  int br;  
  for (br=0;br<255;br++)  
  {  
    analogWrite(5, br);  
    delay(5);           // increase brightness  
  }  
  delay (50);           // delay  
  for (br=255;br>0;br--)  
  {  
    analogWrite(5, br);  
    delay(5);           // decrease brightness  
  }  
}
```



Analog
output
example:
Increase/
decrease LED
brightness

Arduino's PWM frequency is at about 500Hz. The period between the green lines would measure 2 milliseconds each.



[`analogWrite\(\)`](#) is on a scale of 0 – 255.

`analogWrite(255)` means 100% duty cycle (always on).

`analogWrite(127)` is a 50% duty cycle (on half the time).

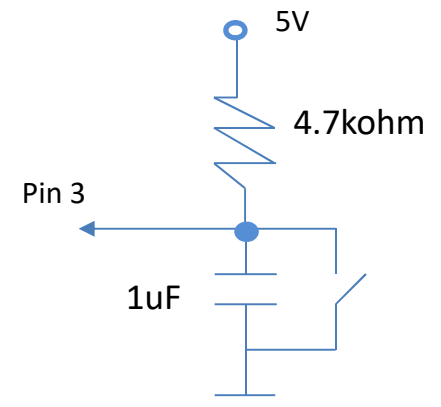
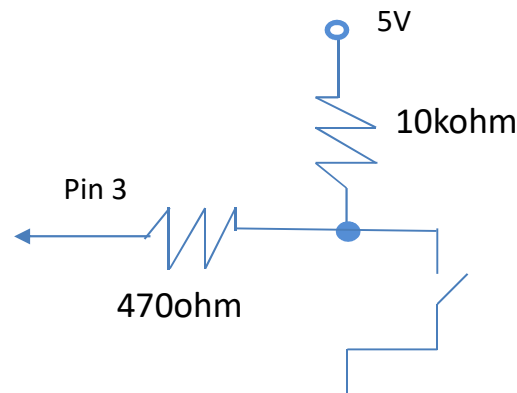
5. Basic digital input

```
void setup() {  
  // initialize digital pin 4 as an output.  
  pinMode(3, INPUT);  
  pinMode(5, OUTPUT);  
}
```

Configure
input pin first

```
void loop() {  
  if (digitalRead (3)==LOW)  
    digitalWrite(5, HIGH);  // turn the LED on  
  else  
    digitalWrite(5, LOW);   // turn the LED off  
  delay(100);               // wait for a while  
}
```

If digital read
is low that is
button
pressed



Use one of these
circuit to interface
switch

6. Digital input using interrupts

	int.0	int.1	int.2	int.3	int.4	int.5
//Board						
//Uno, Ethernet	2	3				
//Mega2560	2	3	21	20	19	18
//Leonardo	3	2	0	1	7	

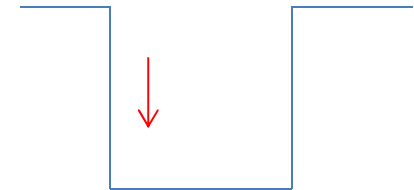
Different interrupt sets

```
void setup() {
  pinMode(3, INPUT);
  pinMode(13, OUTPUT);
  pinMode(5, OUTPUT);
  attachInterrupt(1, Blink, FALLING);
}

void loop() {
  digitalWrite(13, HIGH);    // turn the LED on
  delay (500);
  digitalWrite(13, LOW);    // turn the LED off
  delay(200);
}

void Blink() {
  digitalWrite(5, HIGH);    // turn the LED on
  delay (300);
  digitalWrite(5, LOW);    // turn the LED off
}
```

Attach int 1 (pin 3)
to function "blink"
if there is a fall
detected in signal



Arduino busy
constantly looping
and blinking RED LED

This blink will turn
LED attached to pin
5, only if there is an
interrupt

7. Parsing values to function demo with basic i/o

```
void setup() {  
  // initialize digital pin 4 as an output.  
  pinMode(3, INPUT);  
  pinMode(5, OUTPUT);  
  int number=4;  
  run_my_code(number);  
}
```

Call my
function with
number= 4

```
void run_my_code(int num) {  
  int i;  
  for (i=0;i<num;i++)  
  {  
    digitalWrite(5, HIGH); // turn the LED on  
    delay(300);           // wait  
    digitalWrite(5, LOW);  // turn the LED off  
    delay(300);           // wait  
  }  
}  
  
void loop() { }
```

Describe the function
will take

Infinite loop unused

8. Measuring pulse width

Why unsigned long?

```
unsigned long duration; // duration measured in microseconds
void setup()
{
  Serial.begin (9600);
  pinMode(2, INPUT);
}
```

Initialize serial port.

```
void loop()
{
  duration = pulseIn(2, LOW, 6000000);
  Serial.println(duration/1000); //results in milliseconds
}
```

Measure LOW duration.
Time out after 6 seconds.

Turn on serial monitor at Arduino IDE and observe



Using interrupts

There are two ways a microcontroller responds events

Polling?

Interrupt?



Uno, Nano, Mini,
other 328-based

2, 3

Uno WiFi Rev.2,
Nano Every

all digital **pins**

Mega, Mega2560,
MegaADK

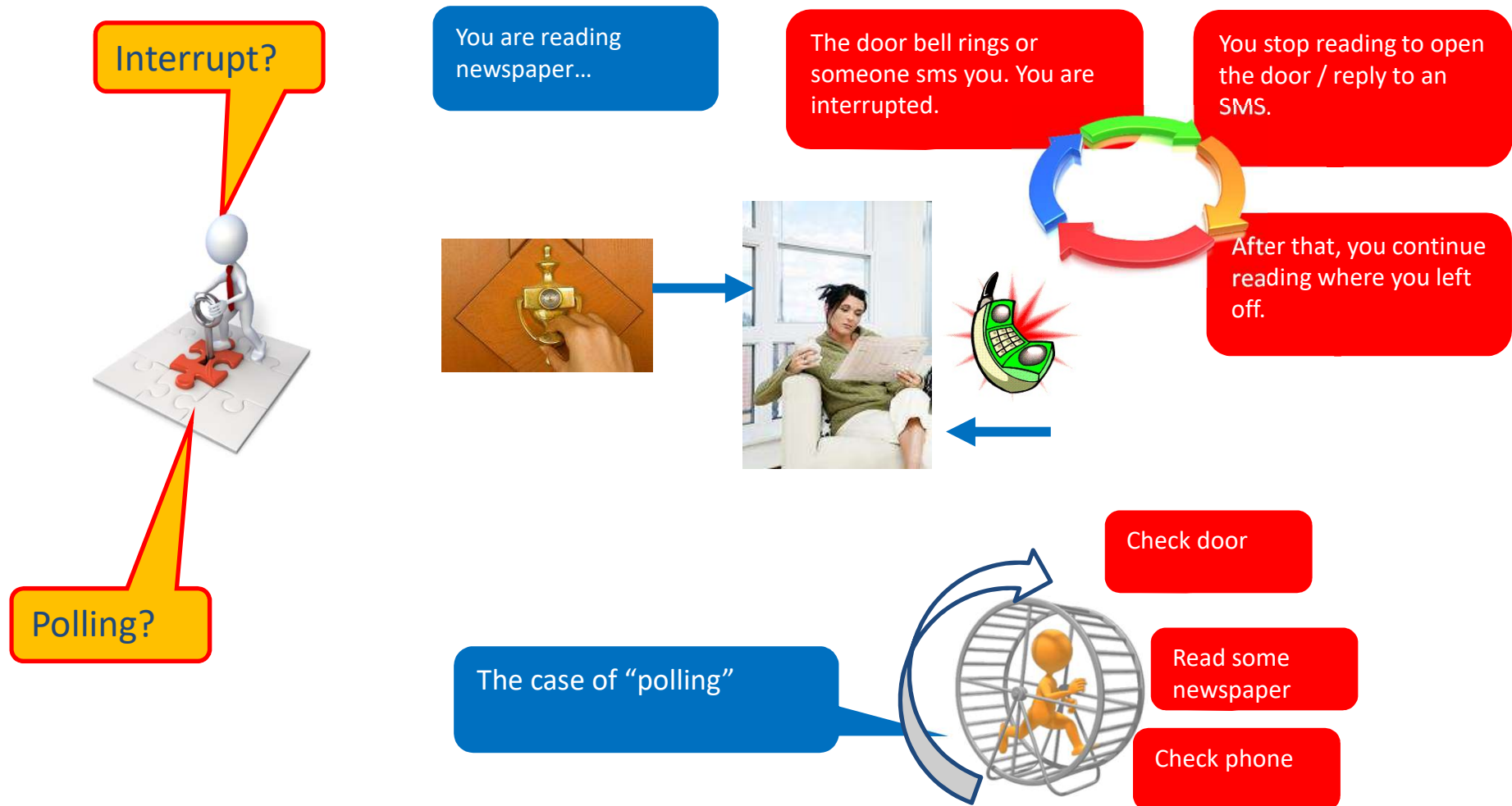
2, 3, 18, 19, 20, 21

Micro, Leonardo,
other 32u4-based

0, 1, 2, 3, 7

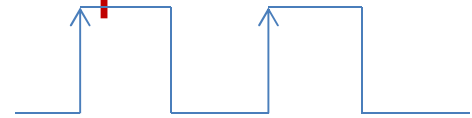
Using interrupts

There are two ways a microcontroller responds events



9.Counting pulses using interrupts

```
volatile int IRQcount;
int pin = 3;
int pin_irq = 0; //IRQ that matches to pin 3
void setup() {
    // put your setup code here, to run once:
    Serial.begin (9600);
    attachInterrupt(pin_irq, IRQcounter, RISING);
}
```



Setup
interrupt

```
void IRQcounter() {
    IRQcount++;
}
```

Interrupt function
will increase counter
every time triggered

```
void loop() {
    cli();//disable interrupts
    IRQcount = 0;
    sei();//enable interrupts
    delay(4000);
    cli();//disable interrupts
    int result = IRQcount;
    sei();//enable interrupts
    Serial.print(F("Counted = "));
    Serial.println(result);
}
```

Disable interrupt and
reset counter

Do nothing for 4
seconds

Send count

10.Find minimum

```
int A[]={13,19,17,8,53};
int value;
int L;
void setup() {
  Serial.begin(9600);
  L=sizeof(A)/sizeof(A[0]);
  value=find_min(A,L);
  Serial.println(value);
  delay (200);
}
```

A quick work
around to find
array size

```
int find_min (int AA[],int L)
{
  int i;
  int mini=9999;
  for (i=0;i<L;i++){
    if (AA[i]<mini) mini=AA[i];
  }
  return mini;
}

void loop() { }
```

11.Find maximum

```
int A[]={13,19,17,22,53};
int value;
int L;
void setup() {
  Serial.begin(9600);
  L=sizeof(A)/sizeof(A[0]);
  value=find_max(A,L);
  Serial.println(value);
}
```

A quick work
around to find
array size

```
int find_max (int AA[],int L)
{
  int i;
  int mx=0;
  for (i=0;i<L;i++){
    if (AA[i]>mx) mx=AA[i];
  }
  return mx;
}

void loop() { }
```

12. Bubble sort and finding median

```
int A[]={13,134,17,22,53};
int value;
int L;
int i;
void setup() {
  Serial.begin(9600);
  L=sizeof(A)/sizeof(A[0]);
  value=find_median(A,L);
  Serial.println(median);
}
```

```
int find_median (int AA[],int L)
{
  int i;
  int change;
  int median;
  int temp;
  while (change !=0){
    change=0;
    for (i=0;i<L-1;i++){
      if (AA[i]>AA[i+1]) {
        temp=AA[i];
        AA[i]=AA[i+1];
        AA[i+1]=temp;
        change=1;
      }
    }
    median=AA[L/2];
  }
  return median;
}

void loop() { }
```

Buffer memory to use
swapping

If current value is
bigger than next one
then swap them

Swap two values

If swap TRUE then
flag change

13. Using switch/case example

```
int value = 0;    // for incoming serial data

void setup() {
    Serial.begin(9600);
}

void loop() {

    // send data only when you receive data:
    if (Serial.available() > 0) {
        // read the incoming byte:
        value = Serial.read();

        // say what you got:
        switch (value){
            case 49:
                Serial.print("I received: 1 ");
                Serial.println(value, DEC);
                break;
            case 50:
                Serial.print("I received: 2 ");
                Serial.println(value, DEC);
                break;
        }
    }
}
```

Depending on the 'value', different actions can be programmed.

This is useful for designing user interface.

14. Code challenge

```
int A[]={13,134,17,22,53};
```

```
...  
...  
...
```

Write a program to sort this array in ascending / descending order and send result to serial monitor



