Robotics

Servo Motor / LCD

School of Computing, Gachon University

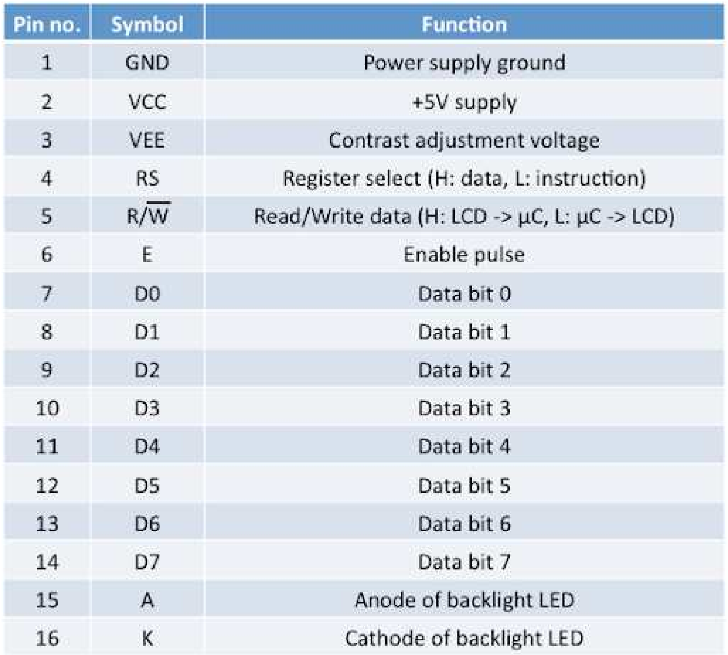
Kang, Sangwoo

Basics

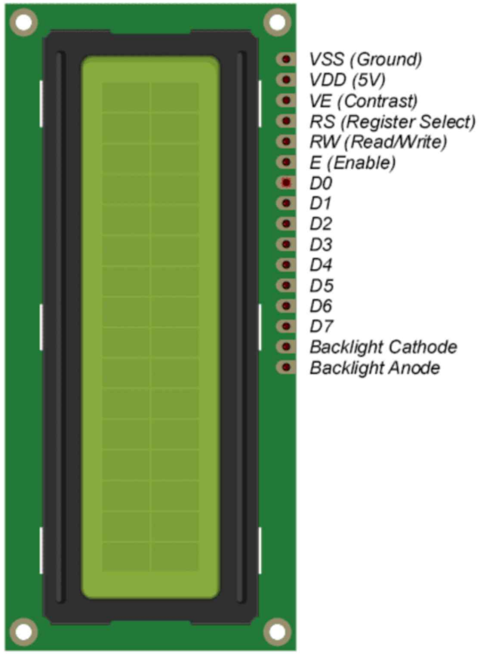
- **Servo Motor**
  - ○ A servomotor is a rotary actuator or linear actuator that allows for precise control of angular or linear position, velocity and acceleration.
  - ○ Servo motors have three wires: power, ground, and signal.
    - – red : 5V
    - – black or brown : GND
    - – yellow, orange or white : signal

Basics [cont'd]

- LCD (Liquid Crystal Display) screen is an electronic display module and find a wide range of applications
  - ○ A 16x2 LCD display is very basic module and is very commonly used in various devices and circuits



Pin no.	Symbol	Function
1	GND	Power supply ground
2	VCC	+5V supply
3	VEE	Contrast adjustment voltage
4	RS	Register select (H: data, L: instruction)
5	R/$\overline{W}$	Read/Write data (H: LCD -> µC, L: µC -> LCD)
6	E	Enable pulse
7	D0	Data bit 0
8	D1	Data bit 1
9	D2	Data bit 2
10	D3	Data bit 3
11	D4	Data bit 4
12	D5	Data bit 5
13	D6	Data bit 6
14	D7	Data bit 7
15	A	Anode of backlight LED
16	K	Cathode of backlight LED



Functions

- **Library : Servo**
  - Allows Arduino/Genuino boards to control a variety of servo motors.
  - To use this library:
    - #include <Servo.h>
- ***Servo object***
  - Define a object of type Servo
  - Syntax
    - Servo *object*;

Functions [cont'd]

- ***object*.attach()**
  - ○ Attach the Servo variable to a pin.
    - The Servo library supports only servos on only two pins: 9 and 10.
  - ○ Syntax
    - *object*.attach(pin)
  - ○ Parameters
    - *object* : a *object* of type Servo
    - pin: the number of the pin that the servo is attached to

Functions [cont'd]

- ***object*.write()**
  - Writes a value to the servo, controlling the shaft accordingly. On a standard servo, this will set the angle of the shaft (in degrees), moving the shaft to that orientation. On a continuous rotation servo, this will set the speed of the servo.
  - Syntax
    - *object*.write(*angle*)
  - Parameters
    - *object* : a object of type Servo
    - *angle*: the value to write to the servo, from 0 to 180

Functions [cont'd]

- Example

```
#include <Servo.h>
Servo myservo;

void setup() {
  myservo.attach(9);
  myservo.write(90);          // set servo to mid-point
}
```

Functions [cont'd]

- **Library : LiquidCrystal**
  - ○ Allows communication with alphanumerical liquid crystal displays (LCDs).
  - ○ To use this library:
    - #include <LiquidCrystal.h>
- **LiquidCrystal()**
  - ○ Creating a variable of type LiquidCrystal
  - ○ Syntax
    - LiquidCrystal *object*(rs, enable, d4, d5, d6, d7)
  - ○ Parameters
    - *object*: a variable of type LiquidCrystal
    - rs: the number of the Arduino pin that is connected to the RS pin on the LCD
    - enable: the number of the Arduino pin that is connected to the enable pin on the LCD
    - The LCD will be controlled using only the four data lines (d4, d5, d6, d7)

Functions [cont'd]

- ***object*.begin()**
  - Initializing the interface to the LCD screen, and specifies the dimensions (width and height) of the display. begin() needs to be called before any other LCD library commands.
  - Syntax
    - *object*.begin(cols, rows)
  - Parameters
    - *object*: a variable of type LiquidCrystal
    - cols: the number of columns that the display has
    - rows: the number of rows that the display has

Functions [cont'd]

- ***object*.print()**
  - Printing text to the LCD
  - Syntax
    - *object*.print(data)
    - *object*.print(data, BASE)
  - Parameters
    - *object*: a variable of type LiquidCrystal
    - data: the data to print (char, byte, int, long, or string)
    - BASE (optional): the base in which to print numbers: BIN for binary (base 2), DEC for decimal (base 10), OCT for octal (base 8), HEX for hexadecimal (base 16)

Functions [cont'd]

- setCursor()
  - Positioning the LCD cursor
    - This function sets the location at which subsequent text written to the LCD will be displayed
  - Syntax
    - *object*.setCursor(col, row)
  - Parameters
    - *object*: a variable of type LiquidCrystal
    - col: the column at which to position the cursor (with 0 being the first column)
    - row: the row at which to position the cursor (with 0 being the first row)

Functions [cont'd]

○ Example

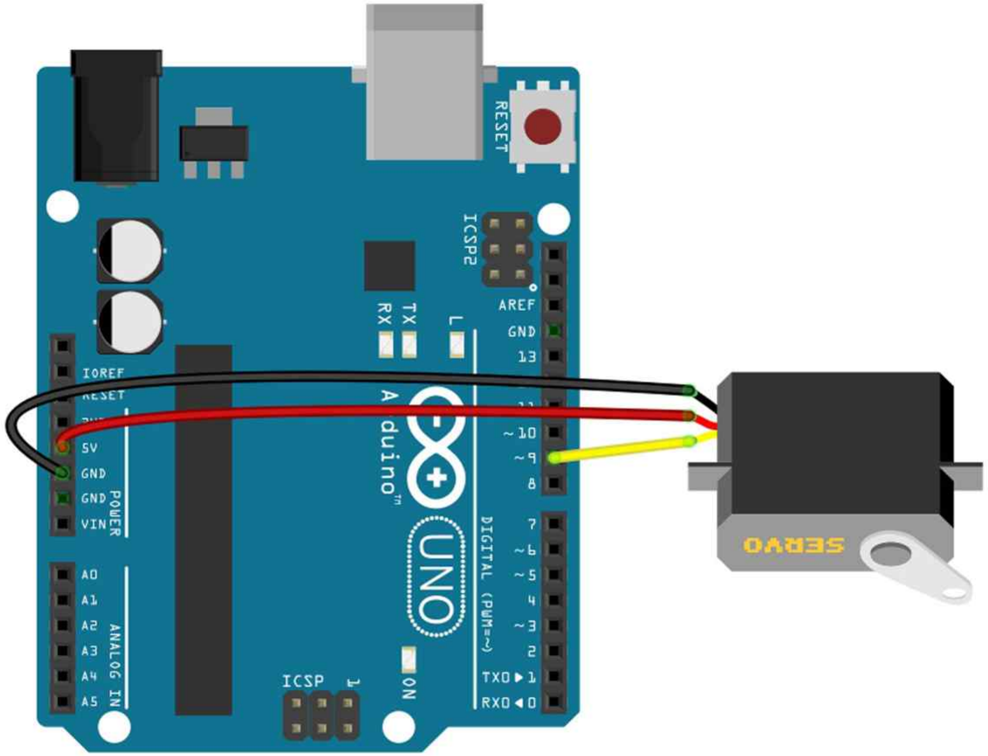
```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);


void setup(){
  lcd.begin(16, 2);                //initialize
  lcd.setCursor(0, 0);
  lcd.print("hello, world!");
}


void loop() {}
```

Functions [cont'd]

- ***object*.noDisplay()**
  - Hiding the LCD cursor.
  - Syntax
    - *object*.noDisplay()
- ***object*.display()**
  - Turning on the LCD display, after it's been turned off with noDisplay().
    - This will restore the text (and cursor) that was on the display.
  - Syntax
    - *object*.display()
- ***object*.clear()**
  - Clearing the LCD screen and positions the cursor in the upper-left corner.
  - Syntax
    - *object*.clear()

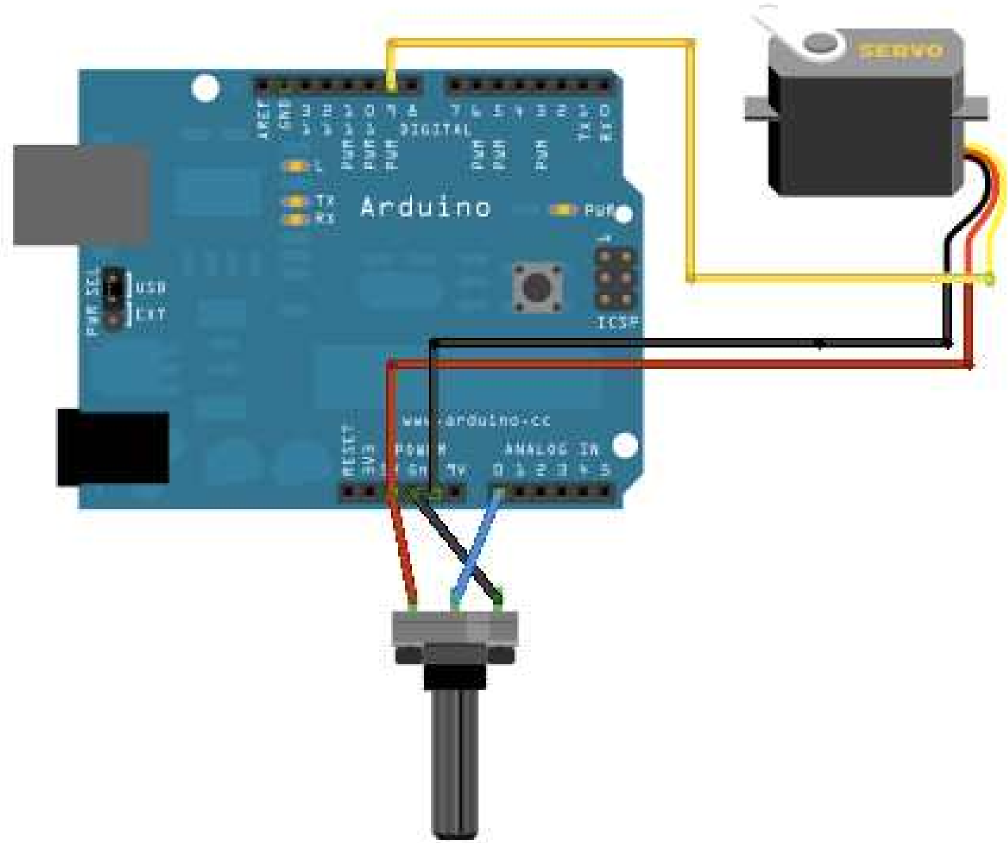
Lab. 1 - Servo motor

- Sweeps the shaft of a servo motor back and forth across 180 degrees.



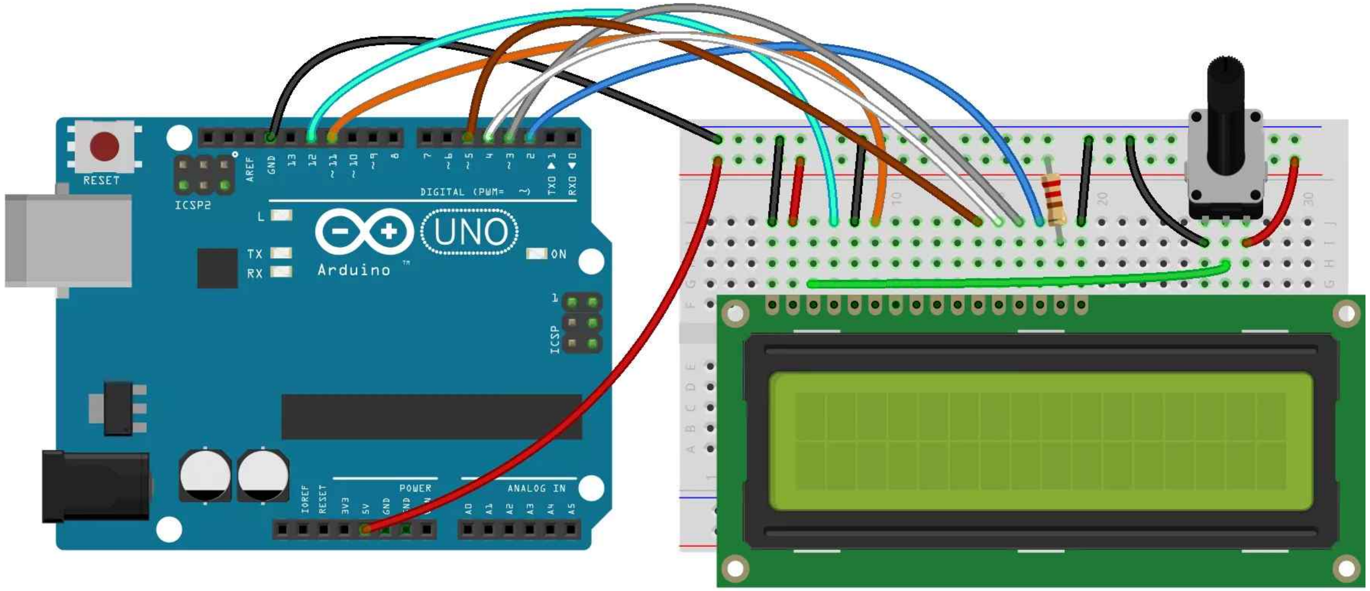
Lab. 2 - Servo motor + Potentiometer

- **Control the position of a servo motor with your Arduino and a potentiometer.**
  - The potentiometer should be wired so that its two outer pins are connected to power (+5V) and ground, and its middle pin is connected to analog input 0 on the board.



Lab. 3 - LCD

- Prints "Hello World!" to the LCD and adjust the screen contrast by The potentiometer.


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX▶1
RX◀0
DIGITAL (PWM= ~)
L
TX
RX
UNO
Arduino
ON
ICSP
IOREF
RESET
3V3
5V
GND
POWER
ANALOG IN
A0
A1
A2
A3
A4
A5
10
20
30