

# 2016 Advanced Arduino

Jonghwa Park

[suakii@gmail.com](mailto:suakii@gmail.com)

GYEONGGI SCIENCE HIGH SCHOOL

# Arduino

Massimo Banzi:

## How Arduino is open-sourcing imagination

TEDGlobal 2012 · 15:46 · Filmed Jun 2012

26 subtitle languages ?

View interactive transcript



Watch later



Favorite



Download



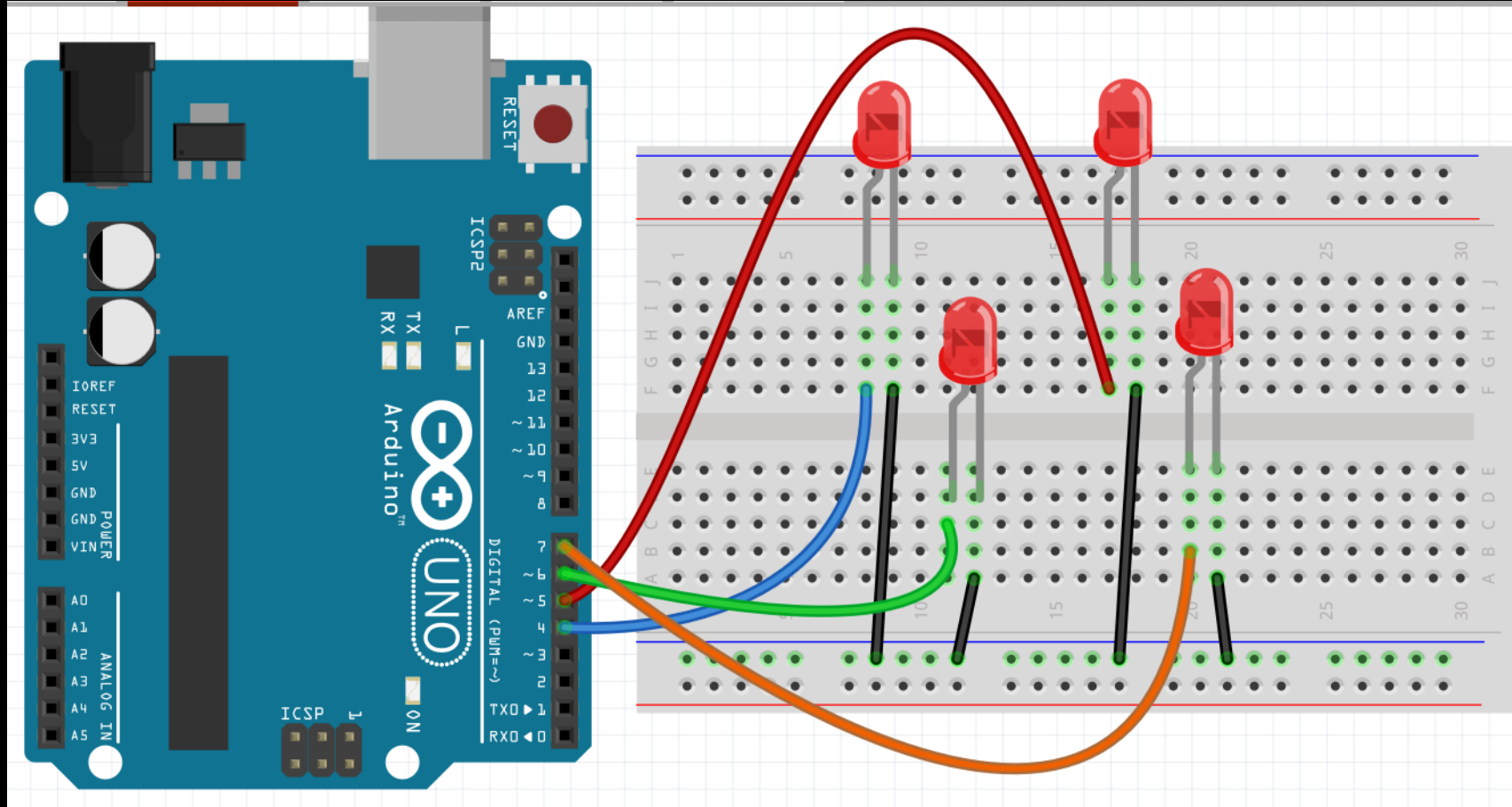
Rate

- [http://www.ted.com/talks/massimo\\_banzi\\_how\\_arduino\\_is\\_open\\_sourcing\\_imagination](http://www.ted.com/talks/massimo_banzi_how_arduino_is_open_sourcing_imagination)

# Massimo Banzi

- Massimo Banzi helped invent the Arduino, a tiny, easy-to-use open-source microcontroller that's inspired thousands of people around the world to make the coolest things they can imagine — from toys to satellite gear. Because, as he says, "You don't need anyone's permission to make something great."

# Serial LED Blink – Hard Ware



# Serial LED Blink – Software

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter1/gs\\_1\\_SerialLEDBlink/gs\\_1\\_SerialLEDBlink.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter1/gs_1_SerialLEDBlink/gs_1_SerialLEDBlink.ino)

# Serial LED Blink –Serial Monitor



# LED Control Using Processing

- First upload this code to Arduino
- [https://github.com/suakii/2016AdArduno/tree/master/Chapter 1/gs 2 LED control arduino](https://github.com/suakii/2016AdArduno/tree/master/Chapter%201/gs%202%20LED%20control%20arduino)

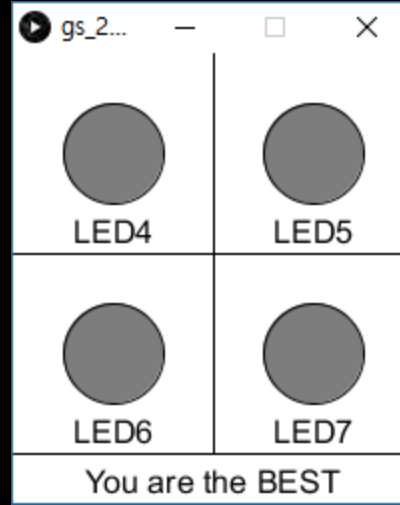
# LED Control Using Processing

- Type this code to Processing
- [https://github.com/suakii/2016AdArduno/blob/master/Chapter1/gs\\_2\\_LED\\_control\\_processing/gs\\_2\\_LED\\_control\\_processing.pde](https://github.com/suakii/2016AdArduno/blob/master/Chapter1/gs_2_LED_control_processing/gs_2_LED_control_processing.pde)



# LED Control Using Processing

- Your are the BEST



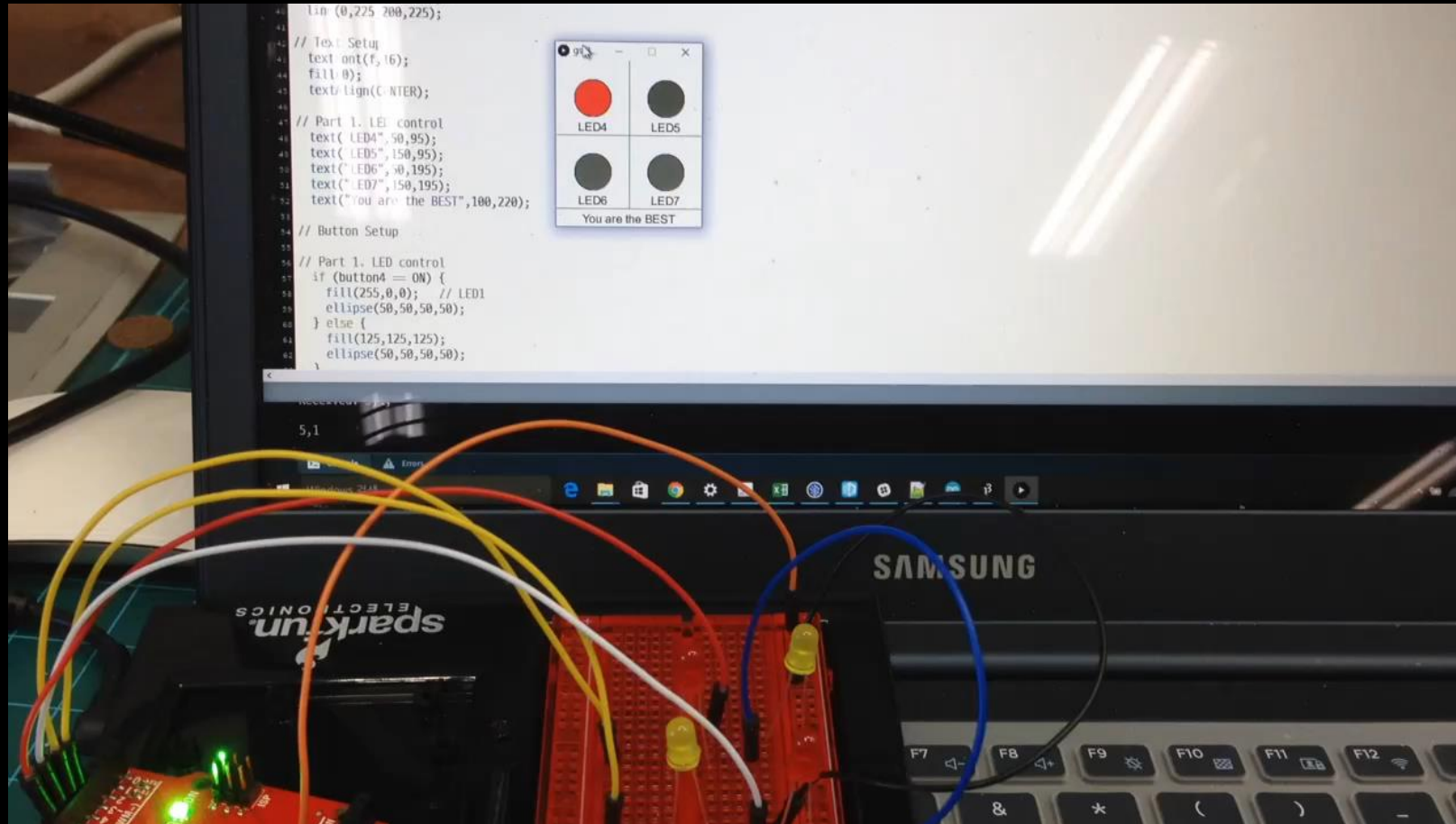
# Reverse : Arduino send to Processing

- Arduino Code
- [https://github.com/suakii/2016AdArduno/blob/master/Chapter1/gs\\_3\\_LED\\_control\\_arduino/gs\\_3\\_LED\\_control\\_arduino.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter1/gs_3_LED_control_arduino/gs_3_LED_control_arduino.ino)

# Reverse : Arduino send to Processing

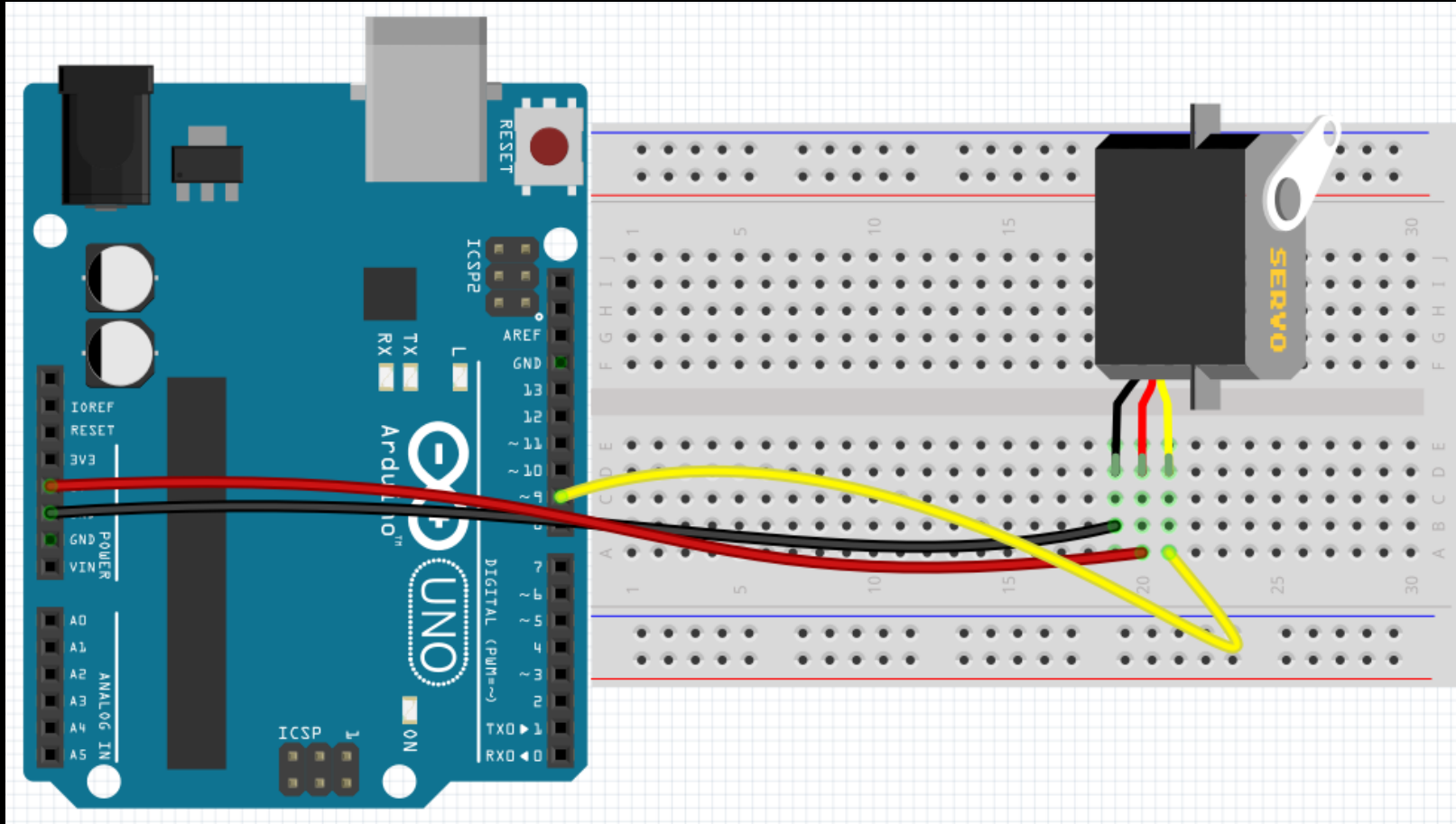
- Processing Code
- [https://github.com/suakii/2016AdArduno/blob/master/Chapter1/gs\\_3\\_LED\\_control\\_processing/gs\\_3\\_LED\\_control\\_processing.pde](https://github.com/suakii/2016AdArduno/blob/master/Chapter1/gs_3_LED_control_processing/gs_3_LED_control_processing.pde)

# Reverse : Arduino send to Processing



Continue Warming UP

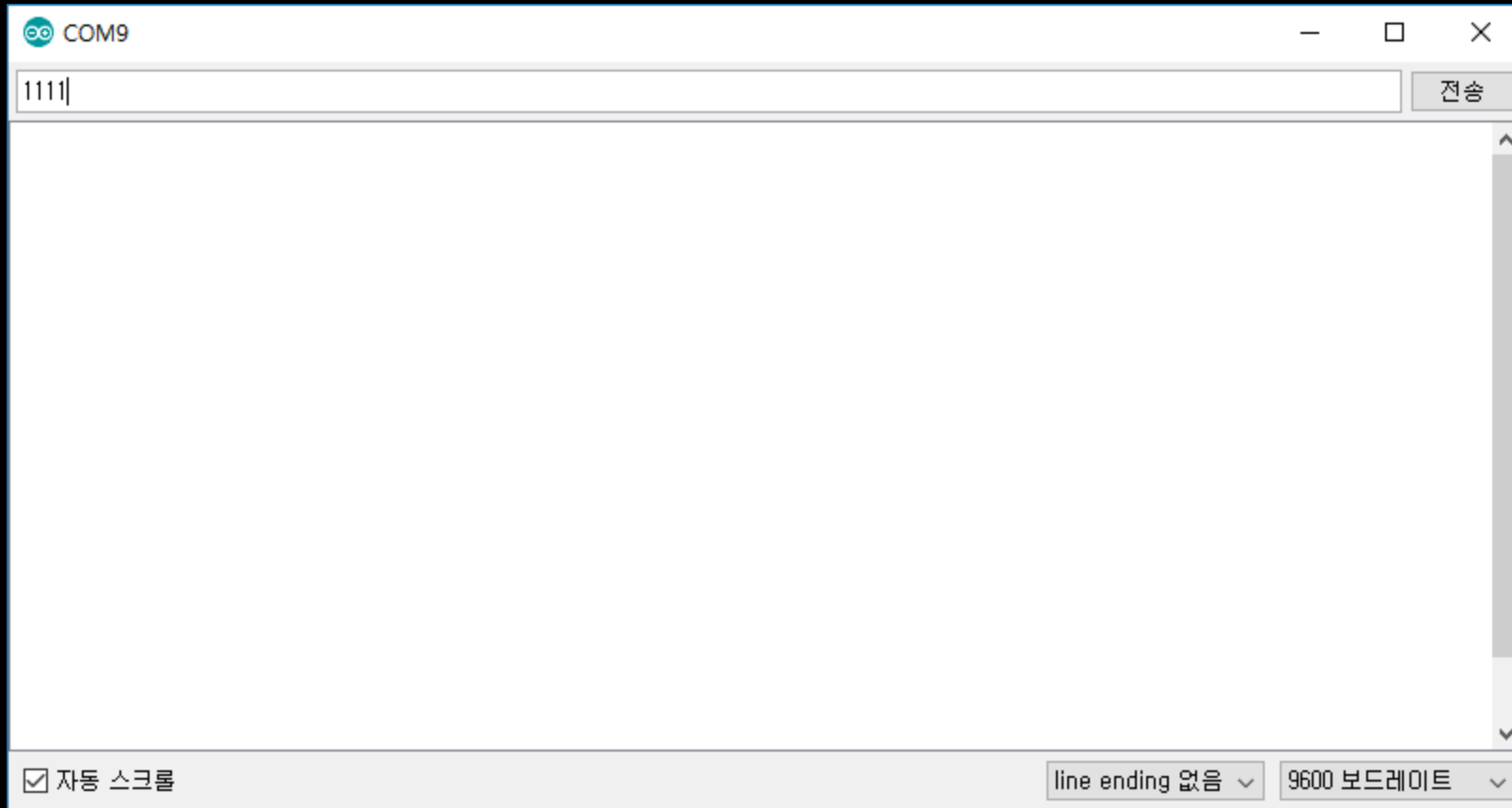
# Servo Test - Simple



# Servo Serial using Serial Monitor

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter2/gs\\_2\\_ServoSerial\\_arduino/gs\\_2\\_ServoSerial\\_arduino.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter2/gs_2_ServoSerial_arduino/gs_2_ServoSerial_arduino.ino)
- //What is code?

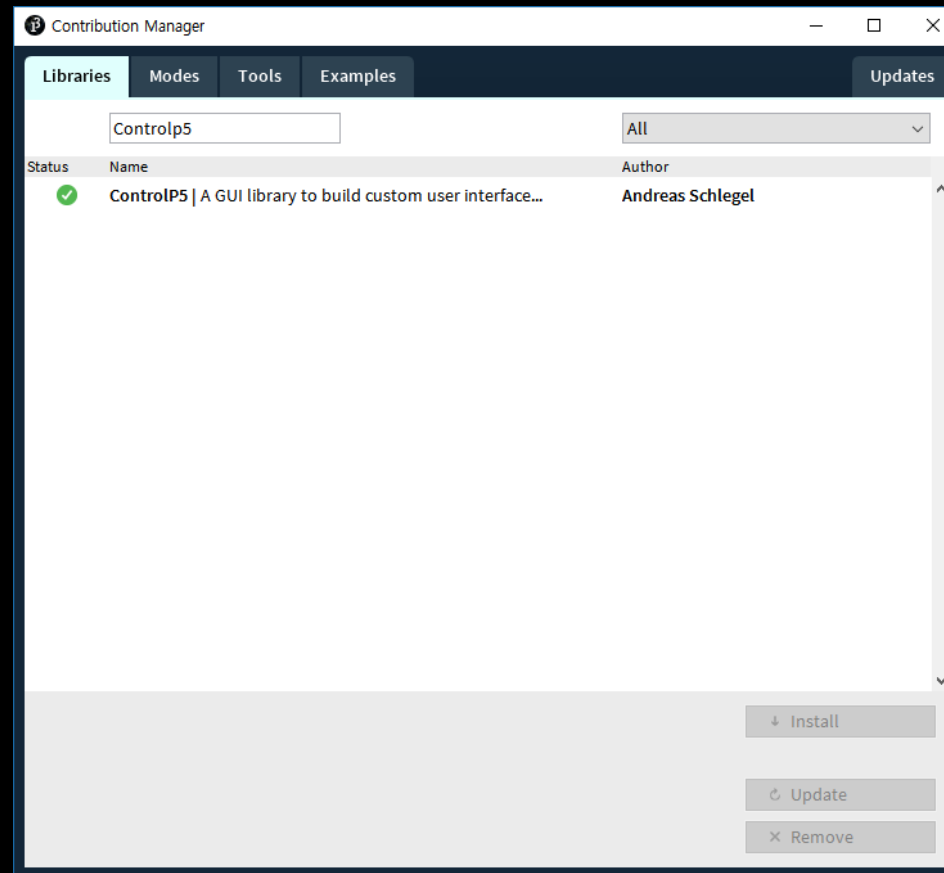
# Send Rotate value to Arduino





# ServoSerial Processing

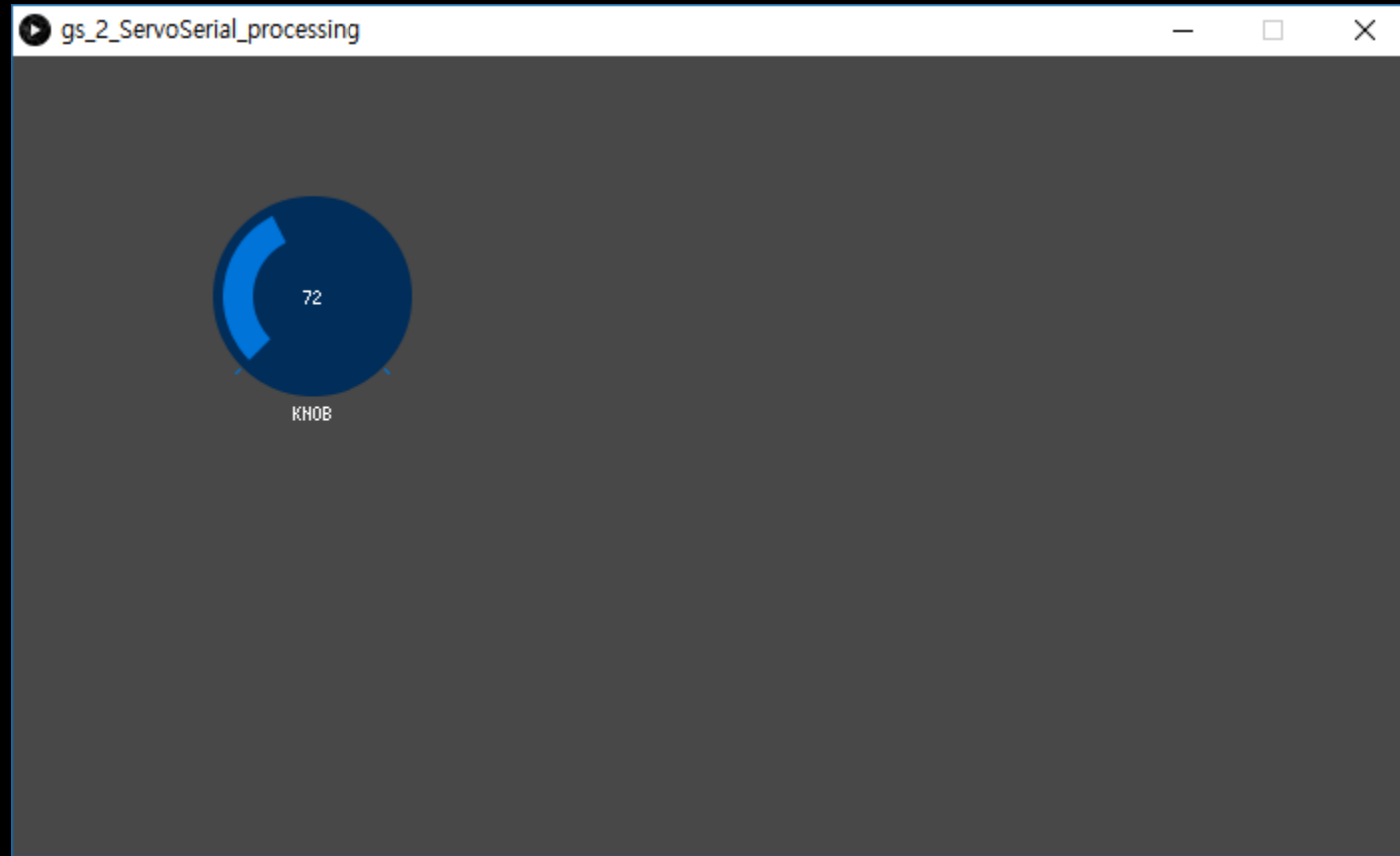
- First Install ControlP5 for UI at Processing



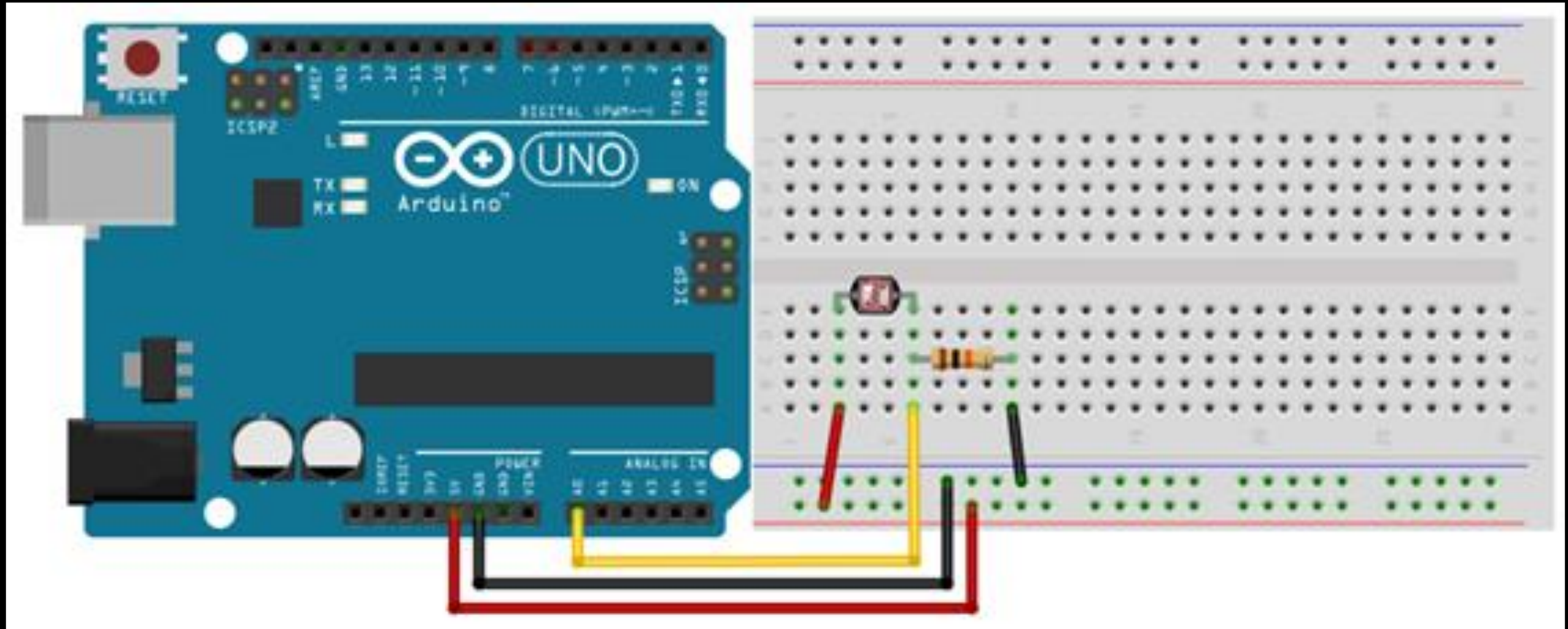
# ServoSerial Processing

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter2/gs\\_2\\_ServoSerial\\_processing/gs\\_2\\_ServoSerial\\_processing.pdf](https://github.com/suakii/2016AdArduno/blob/master/Chapter2/gs_2_ServoSerial_processing/gs_2_ServoSerial_processing.pdf)

# ServoSerial Processing - Cool



# ServoTwoLDR – Light Tracking Simple



# ServoTwoLDR – Light Tracking Simple

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter2/gs\\_4\\_ServoTwoLDR/gs\\_4\\_ServoTwoLDR.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter2/gs_4_ServoTwoLDR/gs_4_ServoTwoLDR.ino)

CU Later....

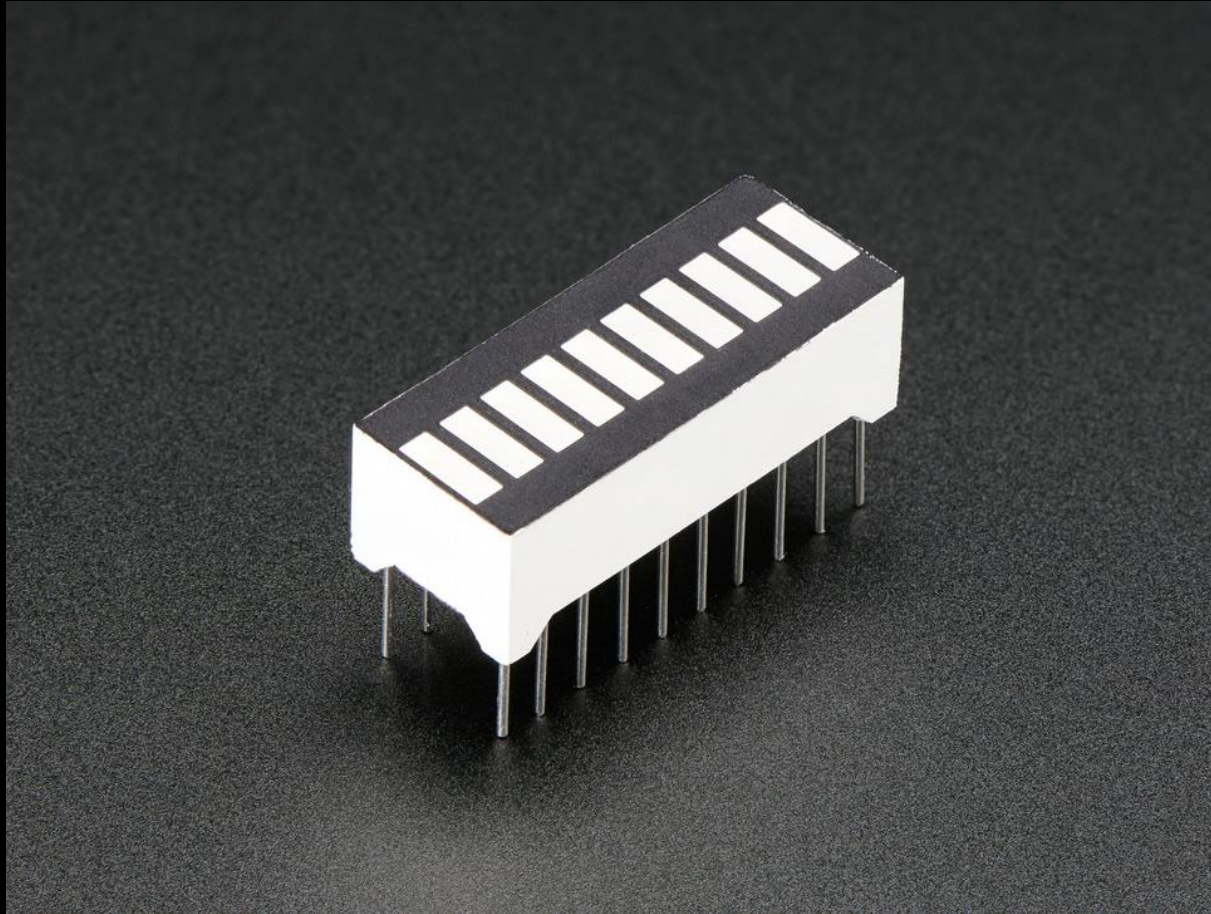
# 2016 Advanced Arduino #2

Jonghwa Park

suakii@gmail.com

GYEONGGI SCIENCE HIGH SCHOOL

# How about using LED Array

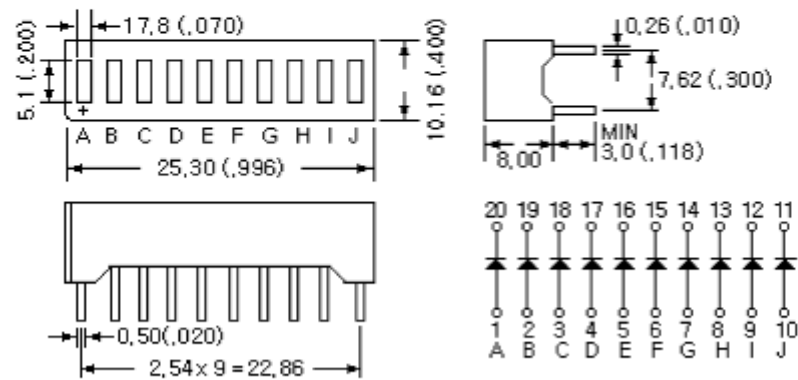




# Simple DataSheet

## FEATURES

1. High brightness
2. Low power consumption; directly drive with I.C
3. Solid state stability; Long operation life
4. Could be jointed two or more units
5. Easily identifiable cathode index



# Using LED Array

- No Code
- Try it Yourself
- [https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs\\_1\\_LedBar/gs\\_1\\_LedBar.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs_1_LedBar/gs_1_LedBar.ino)

# Simple is Best...

- Do u know Shift Register?



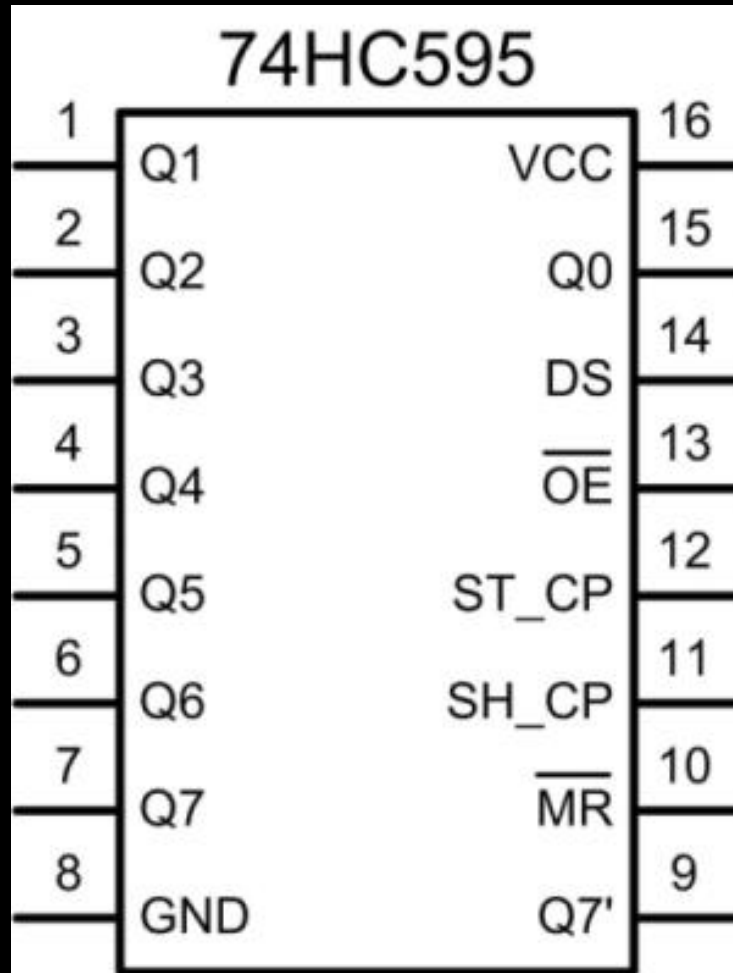
# Shift Register

- We just need 3 pin of Arduino and Shift register to control 8 LED. :)

# 74HC595 Shift Register

The 74HC595; 74HCT595 is an 8-bit serial-in/serial or parallel-out shift register with a storage register and 3-state outputs. Both the shift and storage register have separate clocks. The device features a serial input (DS) and a serial output (Q7S) to enable cascading and an asynchronous reset  $\overline{MR}$  input. A LOW on  $\overline{MR}$  will reset the shift register. Data is shifted on the LOW-to-HIGH transitions of the SHCP input. The data in the shift register is transferred to the storage register on a LOW-to-HIGH transition of the STCP input. If both clocks are connected together, the shift register will always be one clock pulse ahead of the storage register. Data in the storage register appears at the output whenever the output enable input ( $\overline{OE}$ ) is LOW. A HIGH on  $\overline{OE}$  causes the outputs to assume a high-impedance OFF-state. Operation of the  $\overline{OE}$  input does not affect the state of the registers. Inputs include clamp diodes. This enables the use of current limiting resistors to interface inputs to voltages in excess of  $V_{CC}$ .

# 74HC595

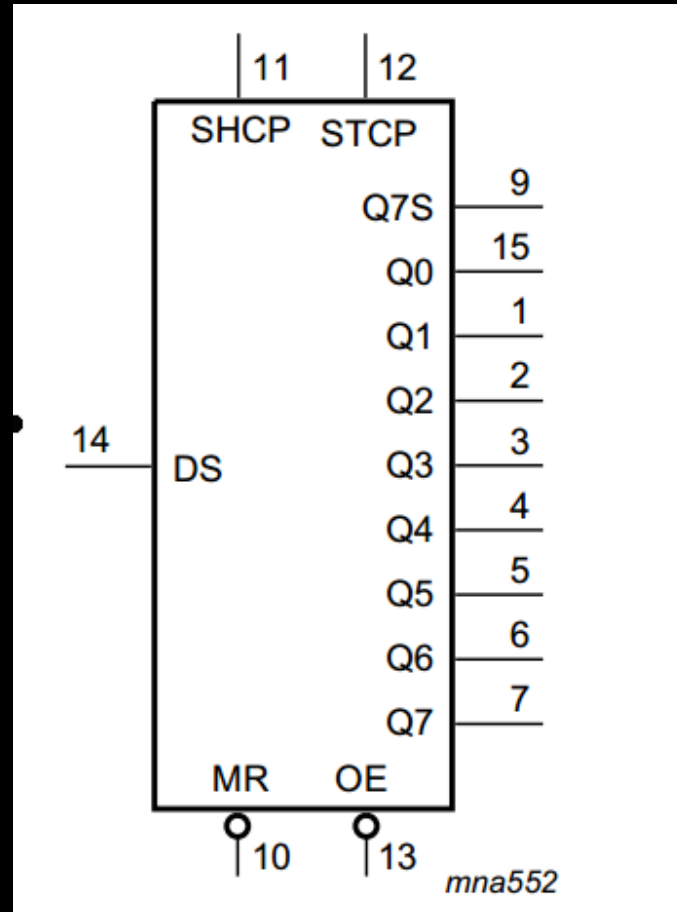


# 74HC595 Pin Mapping

**Table 2. Pin description**

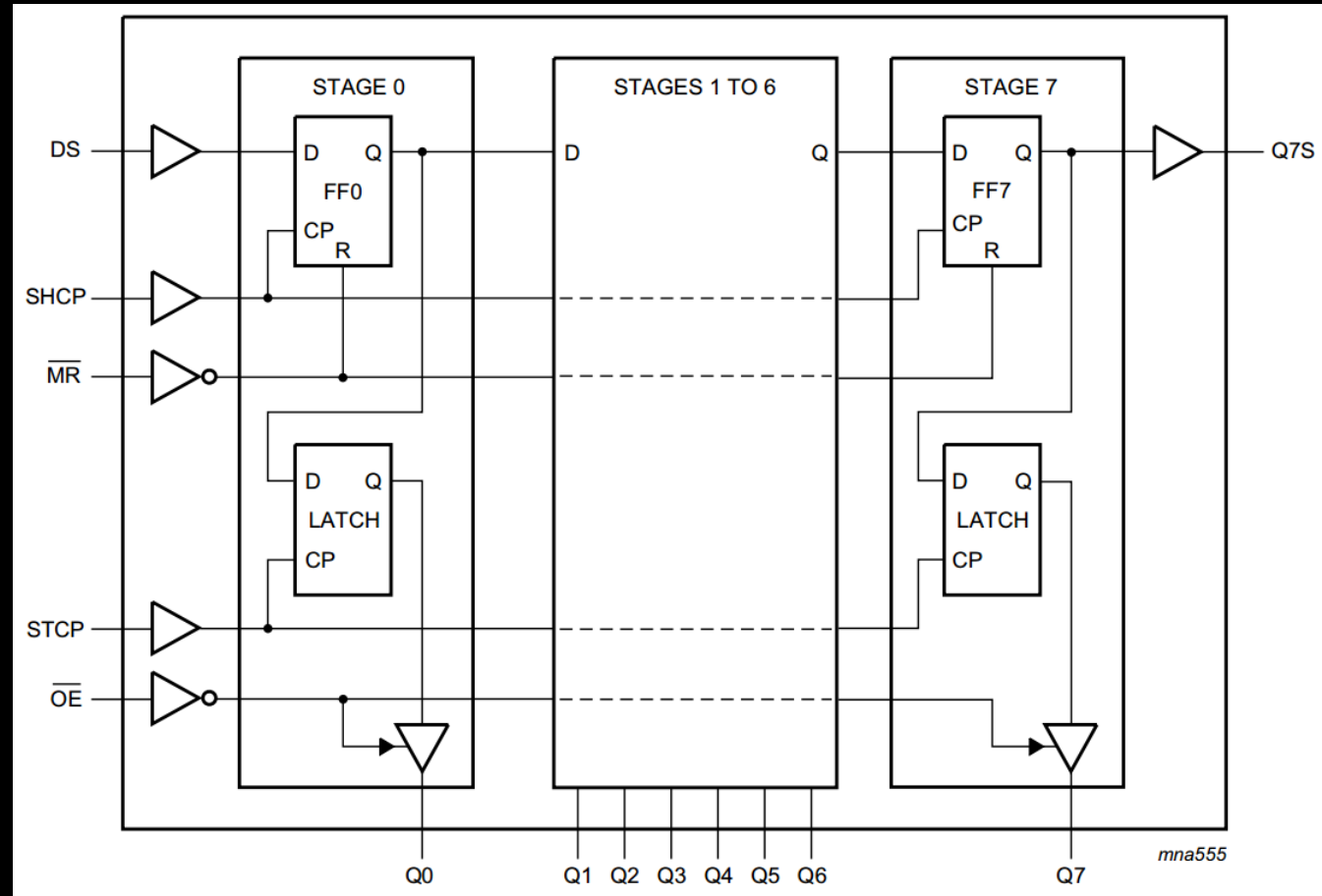
Symbol	Pin	Description
Q0, Q1, Q2, Q3, Q4, Q5, Q6, Q7	15, 1, 2, 3, 4, 5, 6, 7	parallel data output
GND	8	ground (0 V)
Q7S	9	serial data output
$\overline{\text{MR}}$	10	master reset (active LOW)
SHCP	11	shift register clock input
STCP	12	storage register clock input
$\overline{\text{OE}}$	13	output enable input (active LOW)
DS	14	serial data input
Q0	15	parallel data output 0
V <sub>CC</sub>	16	supply voltage

# Pin Mapping





# Inner of Shift Register – Simple

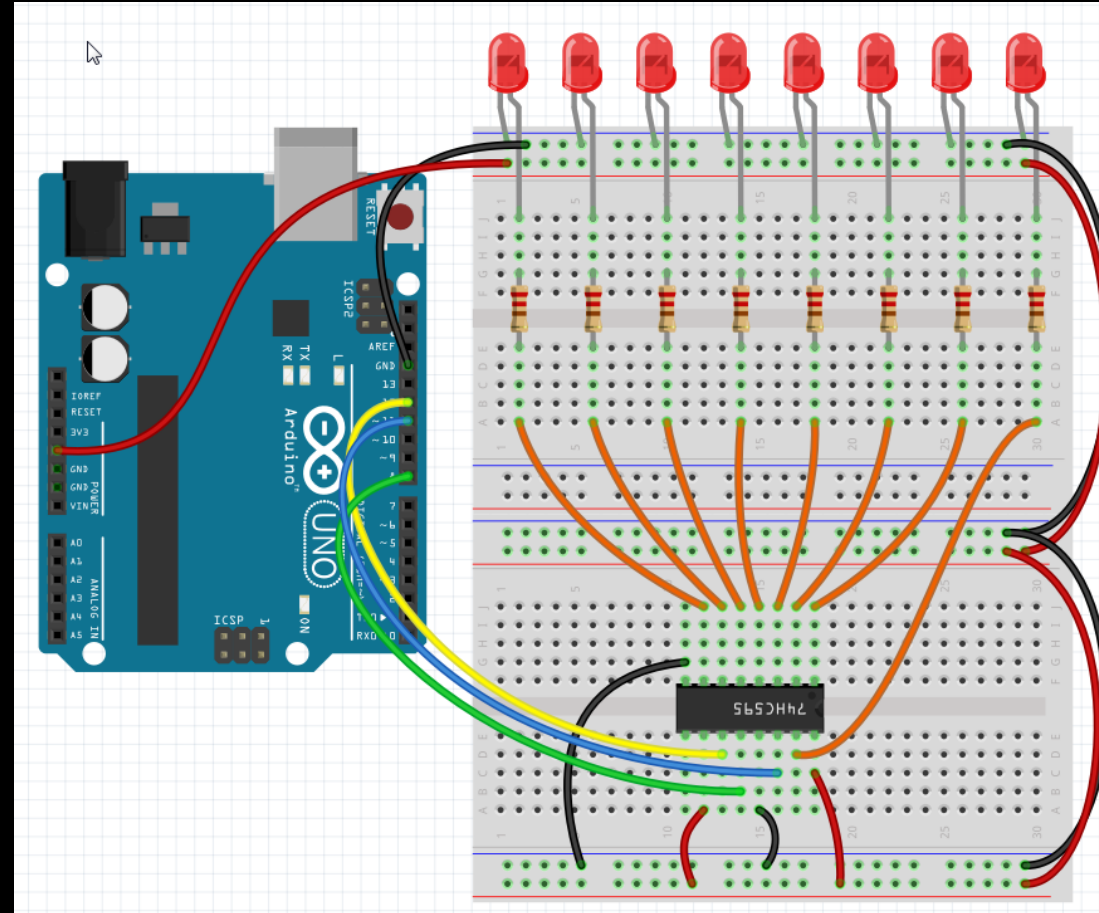


# 74HC595 Function Table – Simple too

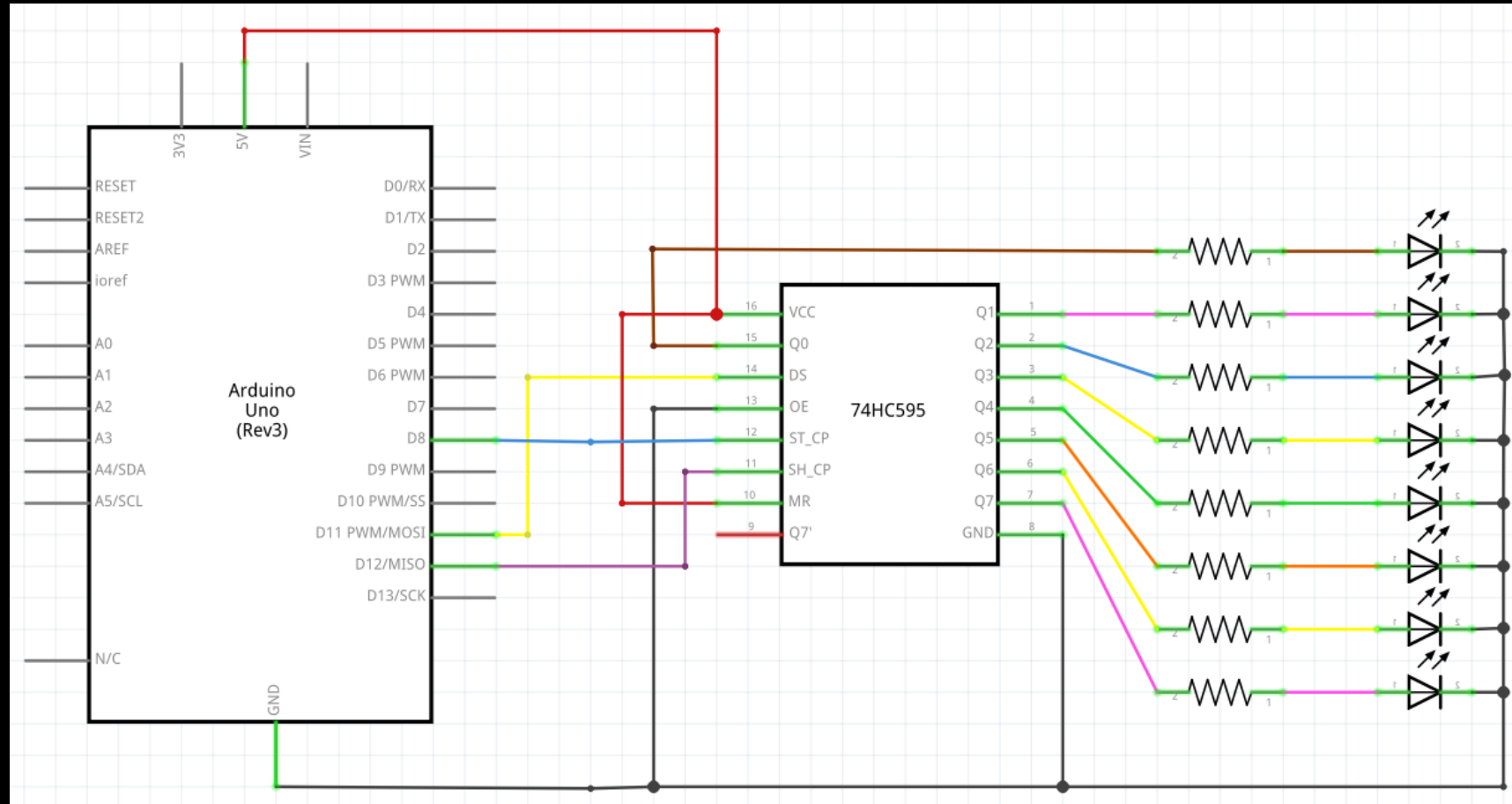
**Table 3.** Function table<sup>[1]</sup>

Control				Input	Output		Function
SHCP	STCP	OE	MR	DS	Q7S	Qn	
X	X	L	L	X	L	NC	a LOW-level on $\overline{\text{MR}}$ only affects the shift registers
X	↑	L	L	X	L	L	empty shift register loaded into storage register
X	X	H	L	X	L	Z	shift register clear; parallel outputs in high-impedance OFF-state
↑	X	L	H	H	Q6S	NC	logic HIGH-level shifted into shift register stage 0. Contents of all shift register stages shifted through, e.g. previous state of stage 6 (internal Q6S) appears on the serial output (Q7S).
X	↑	L	H	X	NC	QnS	contents of shift register stages (internal QnS) are transferred to the storage register and parallel output stages
↑	↑	L	H	X	Q6S	QnS	contents of shift register shifted through; previous contents of the shift register is transferred to the storage register and the parallel output stages

# 74HC595 Connect



# Connect



# Shift Register Test1

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs\\_2\\_1\\_ShiftLegisterTest/gs\\_2\\_1\\_ShiftLegisterTest.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs_2_1_ShiftLegisterTest/gs_2_1_ShiftLegisterTest.ino)

# Shift Register Test2

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs\\_2\\_2\\_ShiftLegisterTest2/gs\\_2\\_2\\_ShiftLegisterTest2.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs_2_2_ShiftLegisterTest2/gs_2_2_ShiftLegisterTest2.ino)

# Shift Register Test3

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs\\_2\\_3\\_ShiftLegisterTest3/gs\\_2\\_3\\_ShiftLegisterTest3.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs_2_3_ShiftLegisterTest3/gs_2_3_ShiftLegisterTest3.ino)

# Shift Register Fade control – OE pin

- To do this, all you need to do, is to change the connection to pin 13 of the 74HC595 so that instead of connecting it to Ground, you connect it to pin 3 of the Arduino.
- [https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs\\_2\\_4\\_ShiftRegisterBrightness/gs\\_2\\_4\\_ShiftRegisterBrightness.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter3/gs_2_4_ShiftRegisterBrightness/gs_2_4_ShiftRegisterBrightness.ino)



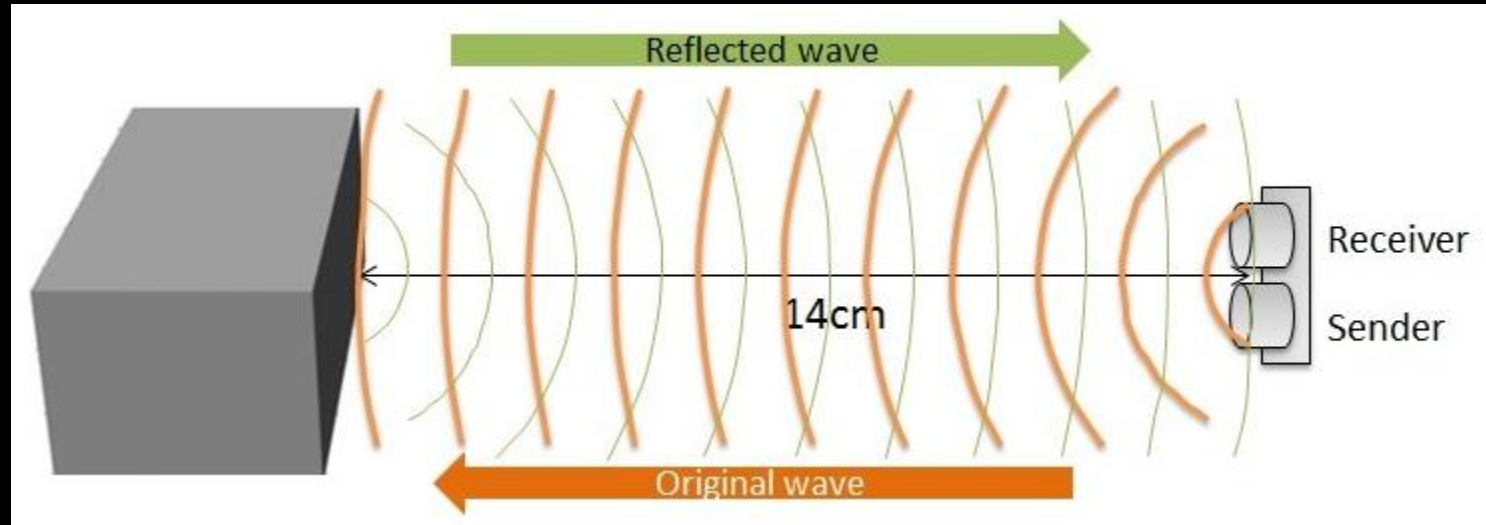
Next

# Ultrasound

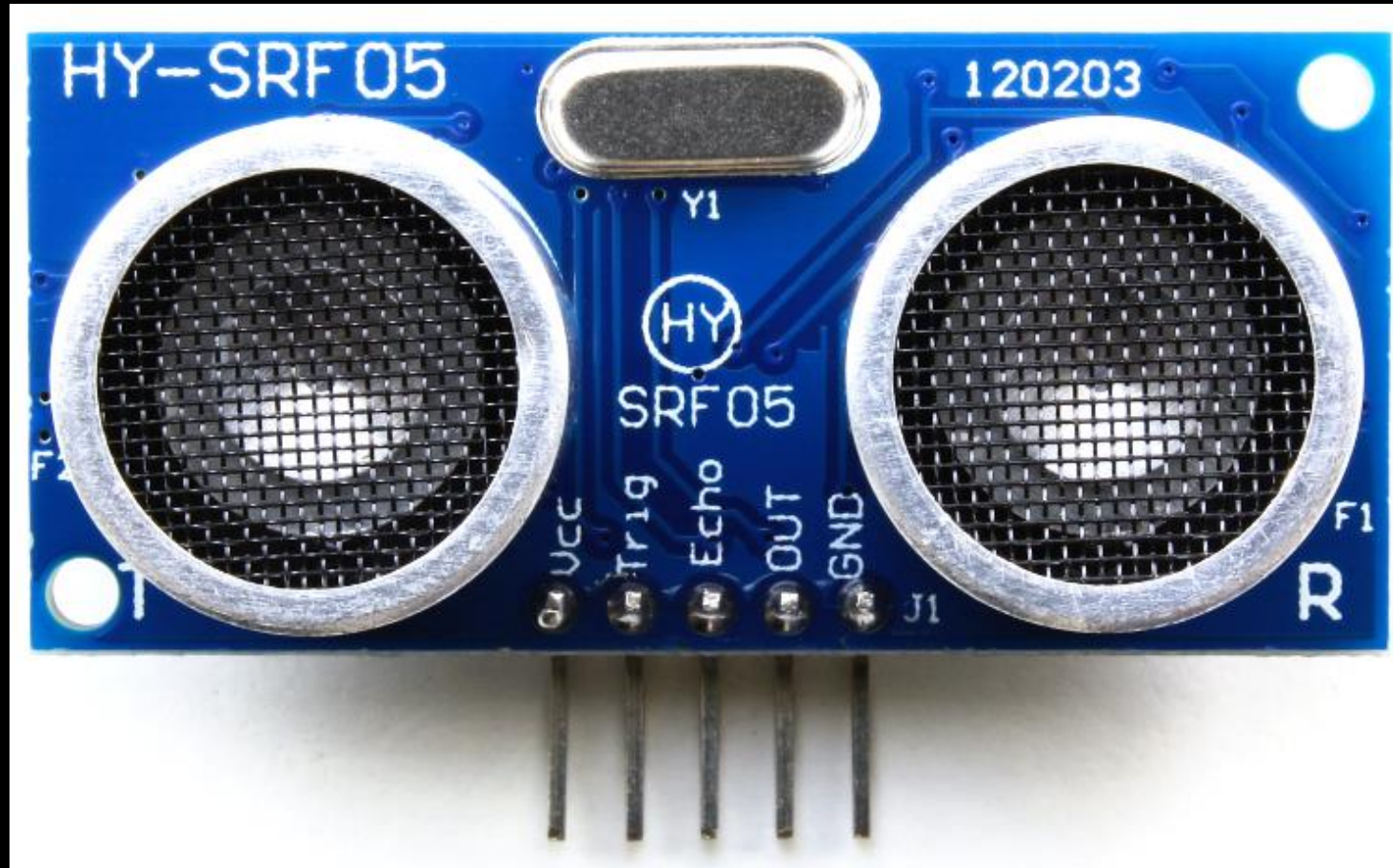
- Galton whistle, one of the first devices to produce ultrasound
- 1893



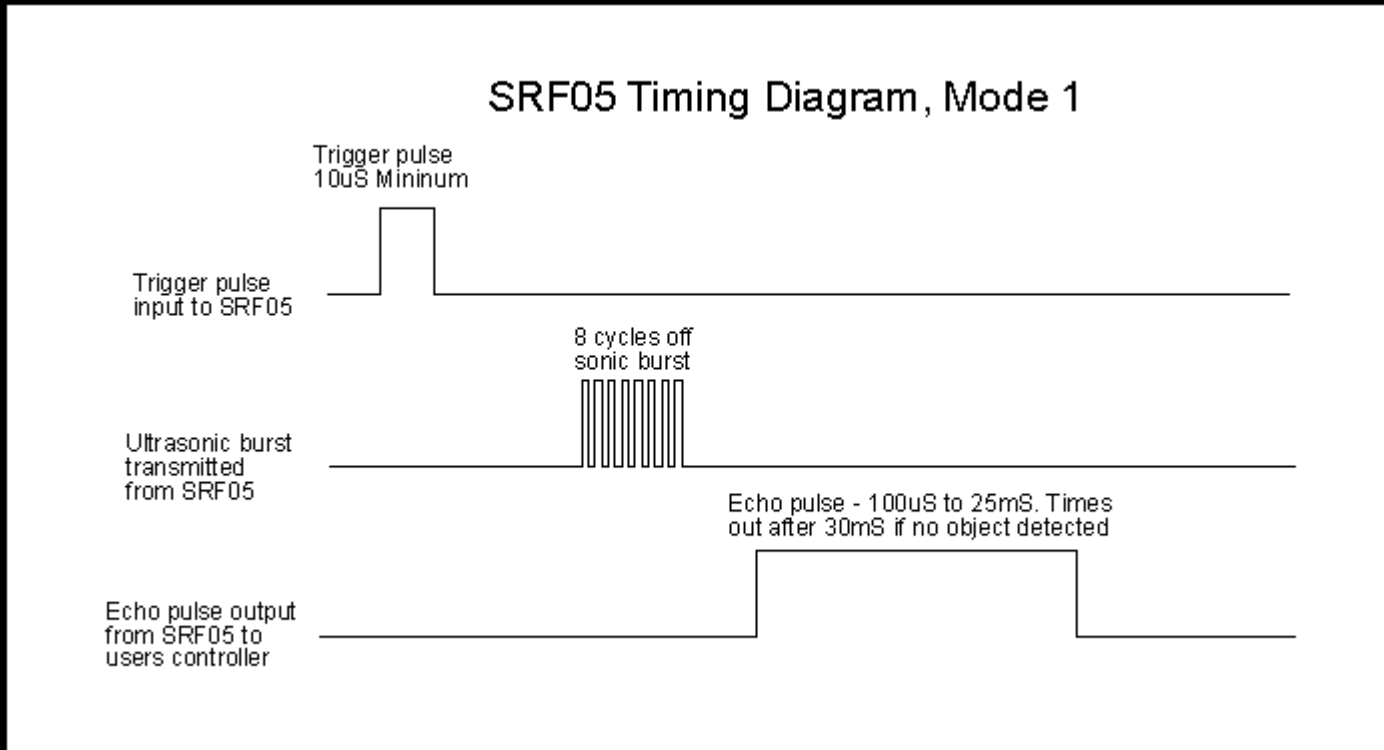
# Ultrasonic Sensor



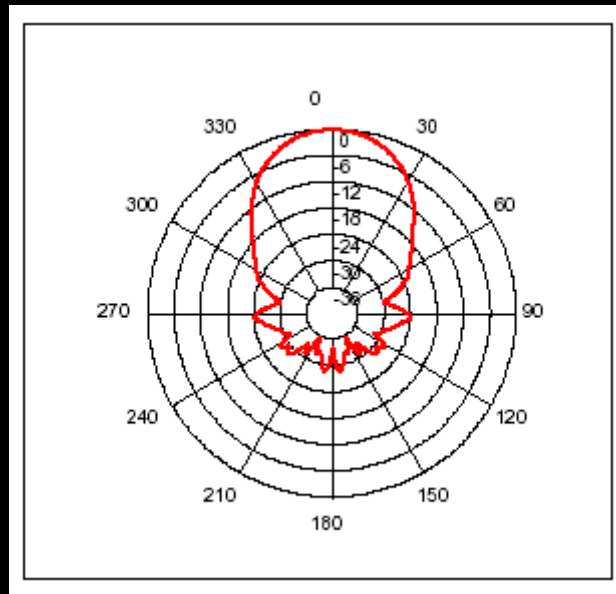
# SRF05



# SRF05 Timing Diagram



# SRF05



# Ultrasonic Rangers FAQ

- [https://www.robot-electronics.co.uk/htm/sonar\\_faq.htm](https://www.robot-electronics.co.uk/htm/sonar_faq.htm)

# SRF05 Test

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter4/gs\\_1\\_SonarTestSimple/gs\\_1\\_SonarTestSimple.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter4/gs_1_SonarTestSimple/gs_1_SonarTestSimple.ino)



# SRF05 Test with Processing

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter4/gs\\_2\\_SonarGraph\\_processing/gs\\_2\\_SonarGraph\\_processing.pde](https://github.com/suakii/2016AdArduno/blob/master/Chapter4/gs_2_SonarGraph_processing/gs_2_SonarGraph_processing.pde)

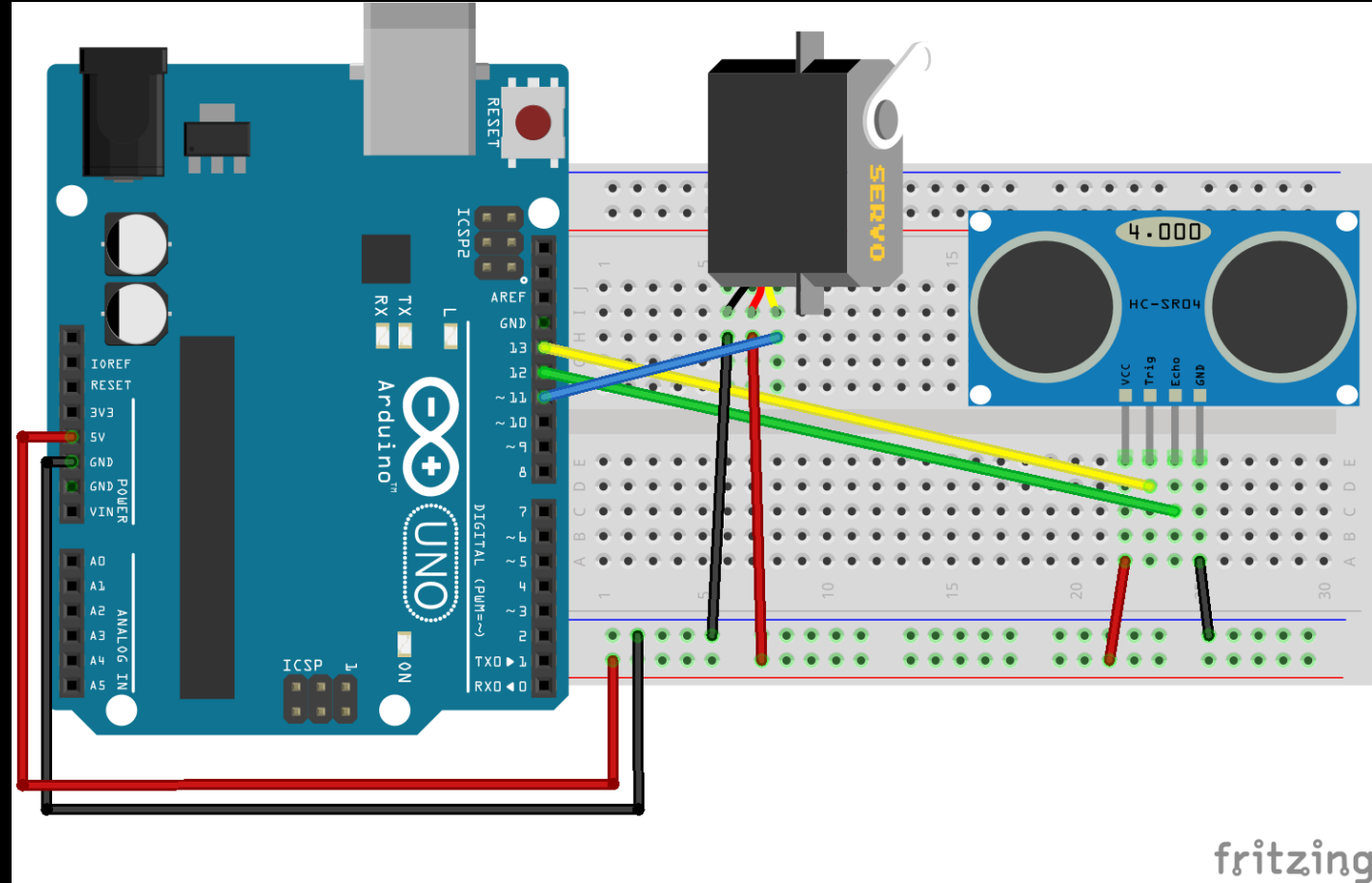
# SRF05 with LED Bar

- Implement Yourself

# Sonar with Piezzo – Simple Sound

- [https://github.com/suakii/2016AdArduno/tree/master/Chapter4/gs\\_3\\_SonarMusic](https://github.com/suakii/2016AdArduno/tree/master/Chapter4/gs_3_SonarMusic)

# Sonar with Servo



# Sonar Radar Arduino

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter4/gs\\_4\\_SonarRadar\\_arduino/gsonar4\\_SonarRadar\\_arduino.ino](https://github.com/suakii/2016AdArduno/blob/master/Chapter4/gs_4_SonarRadar_arduino/gsonar4_SonarRadar_arduino.ino)

# Sonar Radar Processing

- [https://github.com/suakii/2016AdArduno/blob/master/Chapter4/gs\\_4\\_SonarRadar\\_processing/gs\\_4\\_SonarRadar\\_processing.pdf](https://github.com/suakii/2016AdArduno/blob/master/Chapter4/gs_4_SonarRadar_processing/gs_4_SonarRadar_processing.pdf)

CU Later....