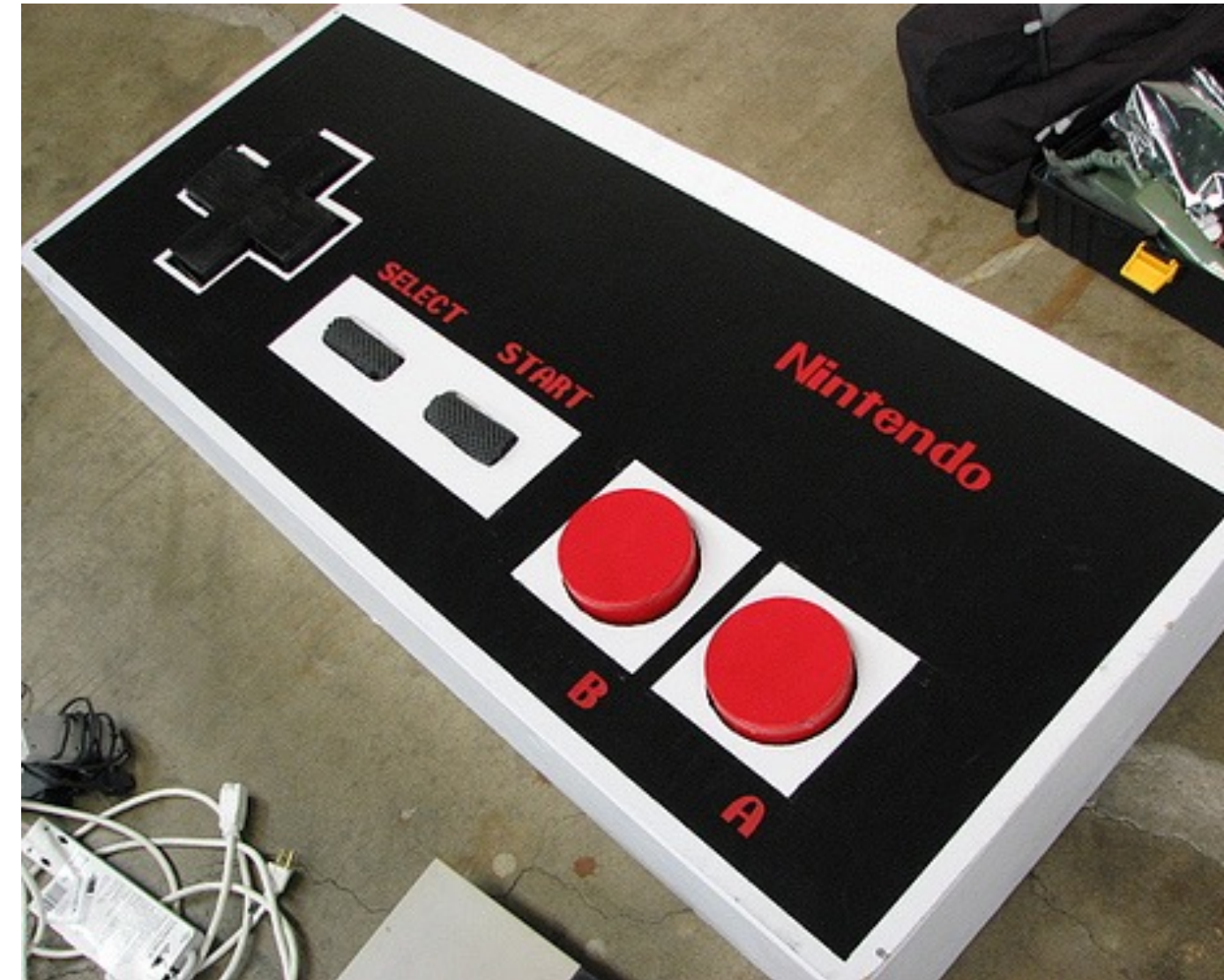
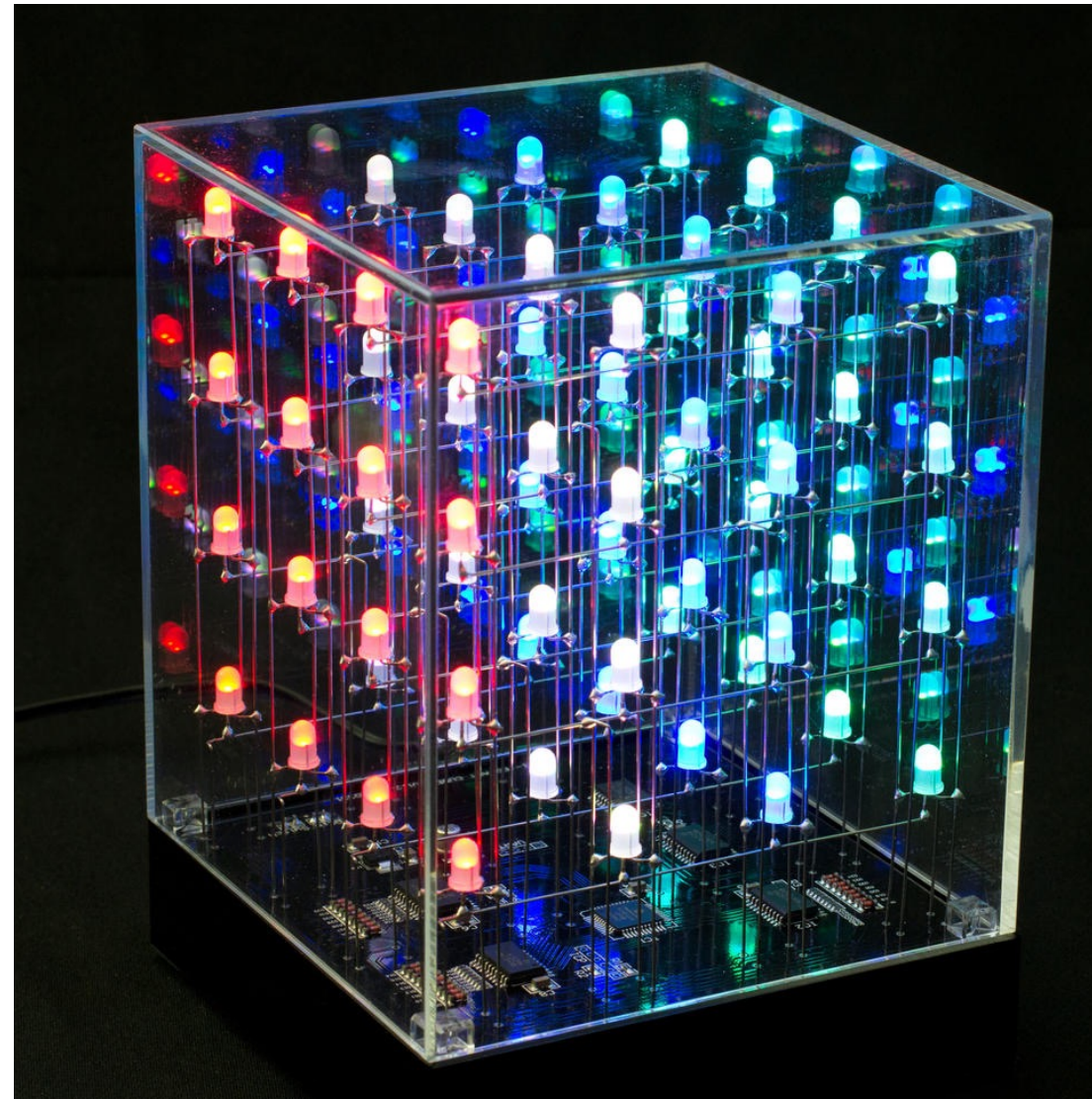


Shift Register

Huaishu Peng | UMD CS | Fall 2024



Shift Register - Why



Shift Register - How

SIPO Vs PISO Shift Registers

Shift registers come in two basic types, either SIPO (Serial-In-Parallel-Out) or PISO (Parallel-In-Serial-Out). The popular SIPO chip is 74HC595, and the PISO chip is 74HC165.

The first type, SIPO, is useful for controlling a large number of outputs, like LEDs. While the latter type, PISO, is good for gathering a large number of inputs, like buttons

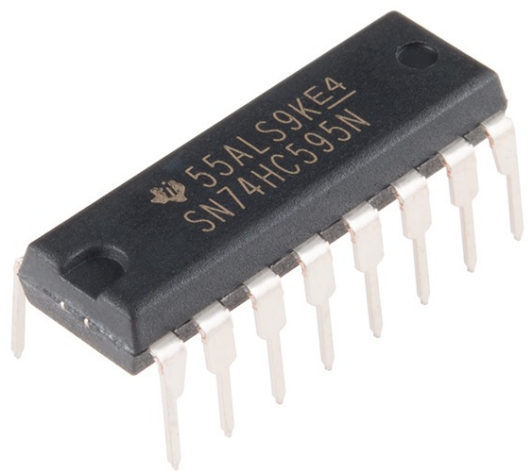


Shift Register - How – 74HC595

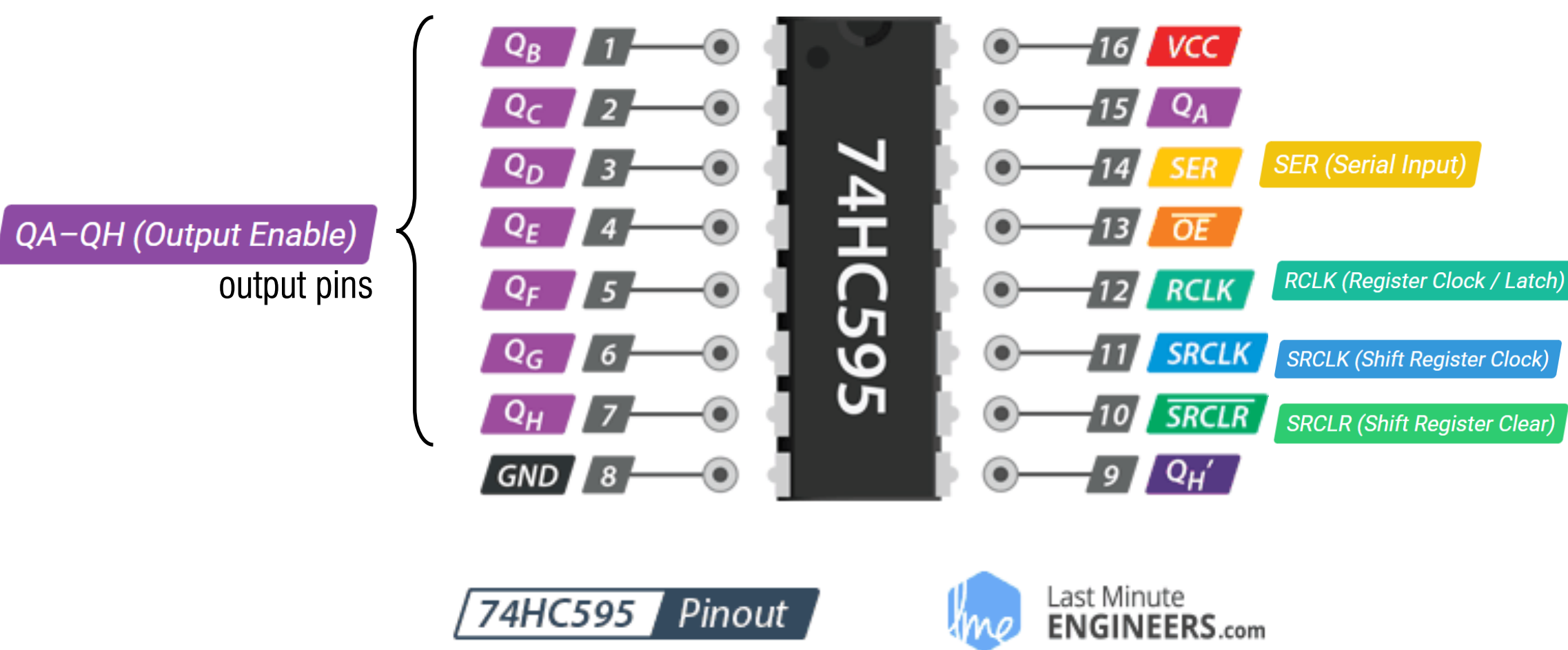
SIPO Vs PISO Shift Registers

Shift registers come in two basic types, either SIPO (Serial-In-Parallel-Out) or PISO (Parallel-In-Serial-Out). The popular SIPO chip is 74HC595, and the PISO chip is 74HC165.

The first type, SIPO, is useful for controlling a large number of outputs, like LEDs. While the latter type, PISO, is good for gathering a large number of inputs, like buttons



Shift Register - How – 74HC595



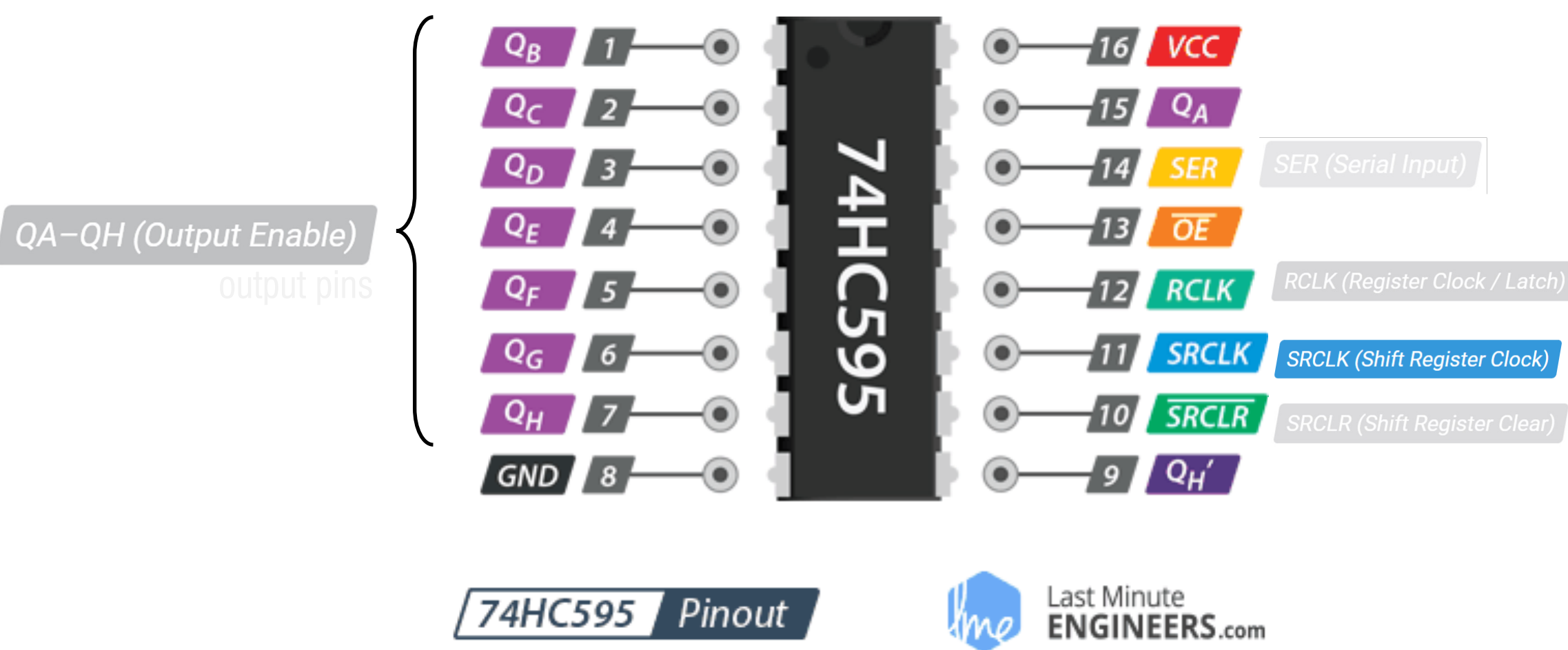
to feed data into the shift register a bit at a time

when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register

the clock for the shift register

allows us to reset the entire Shift Register to 0

Shift Register - How – 74HC595



to feed data into the shift register a bit at a time

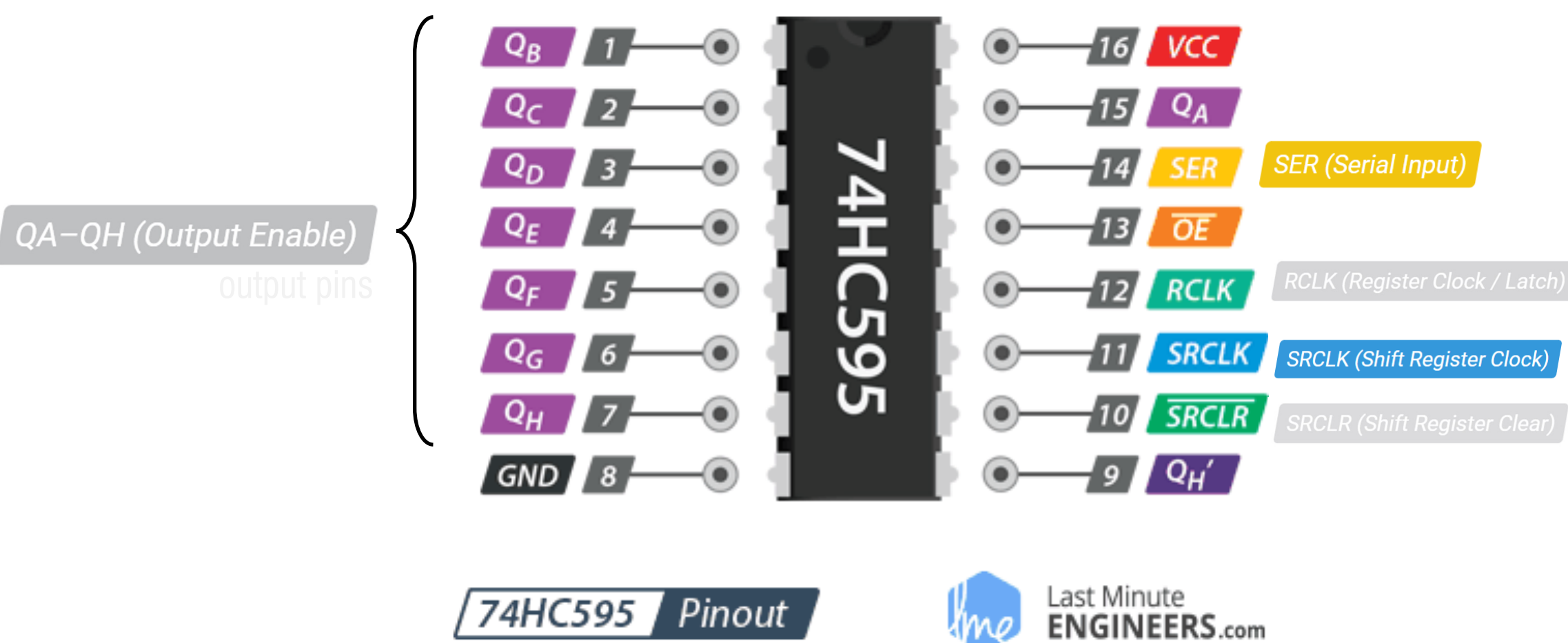
when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register

the clock for the shift register

allows us to reset the entire Shift Register to 0



Shift Register - How – 74HC595



to feed data into the shift register a bit at a time (0)

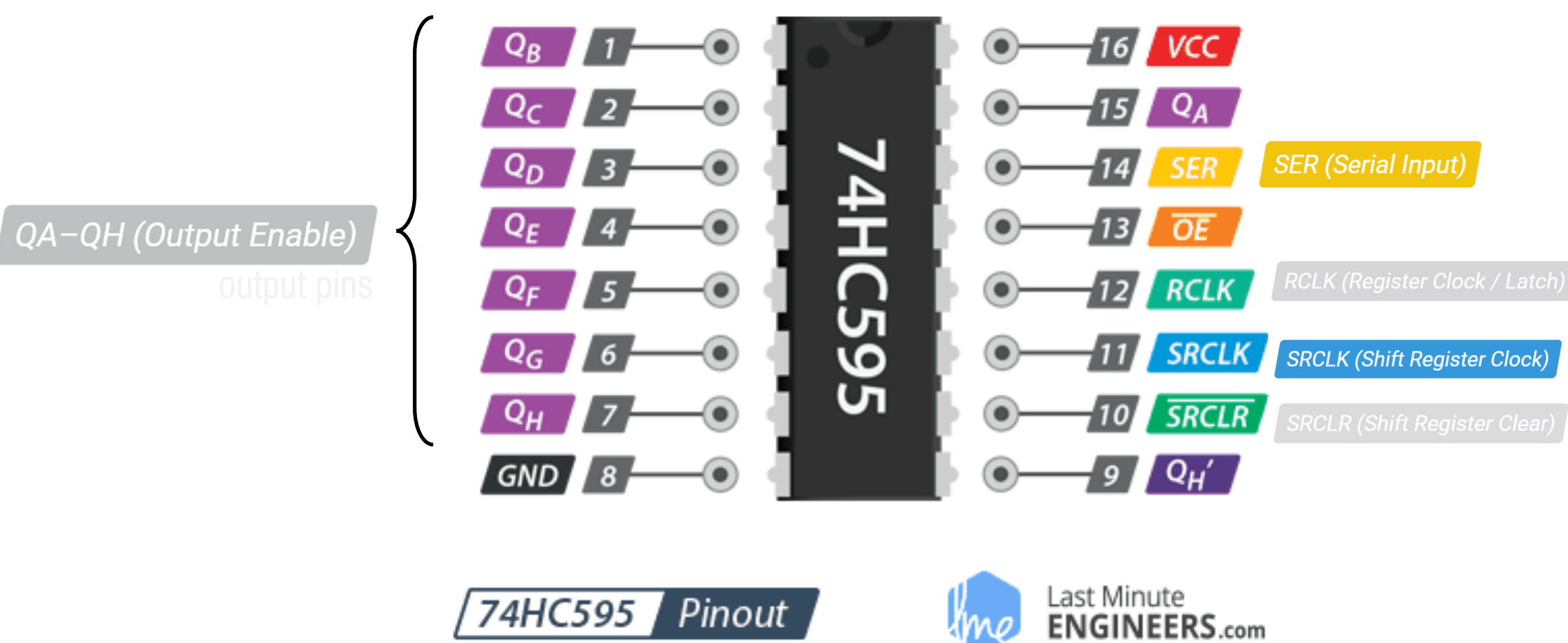
when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register

the clock for the shift register

allows us to reset the entire Shift Register to 0



Shift Register - How – 74HC595



to feed data into the shift register a bit at a time (0)

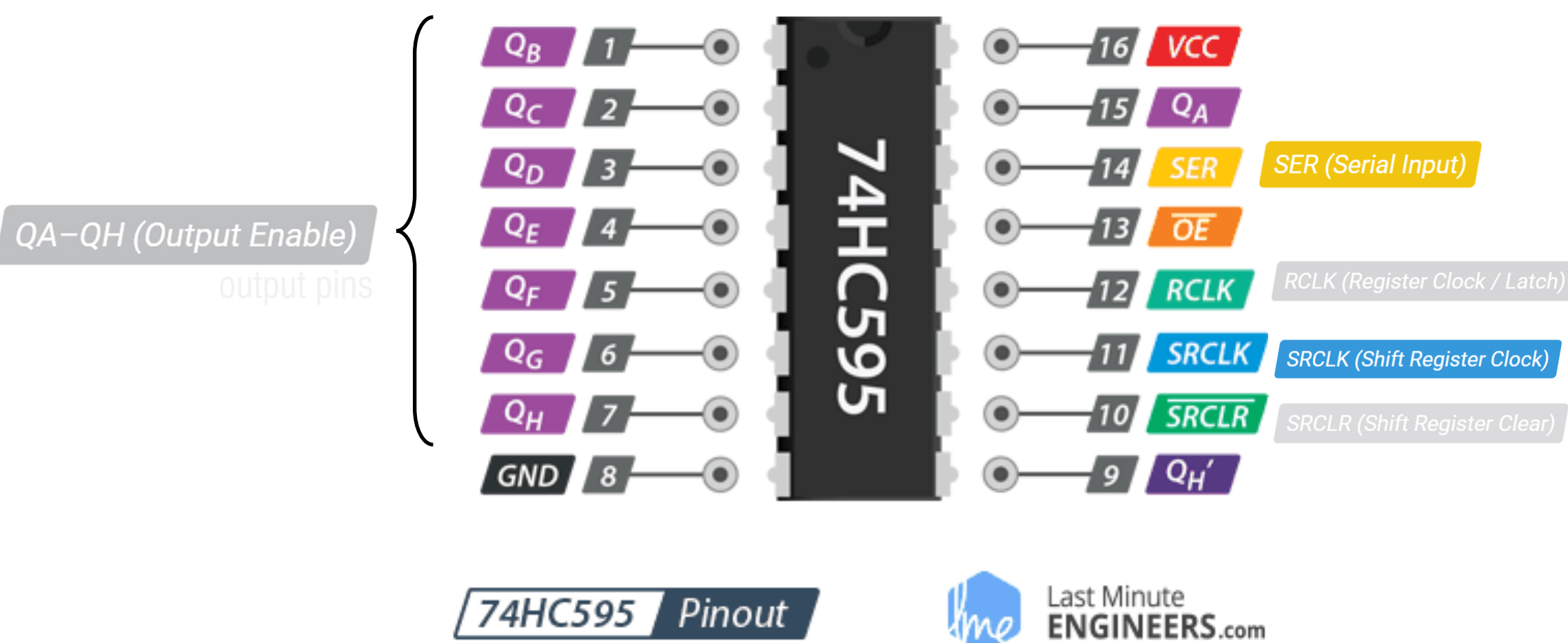
when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register

the clock for the shift register

allows us to reset the entire Shift Register to 0



Shift Register - How – 74HC595



to feed data into the shift register a bit at a time (0)

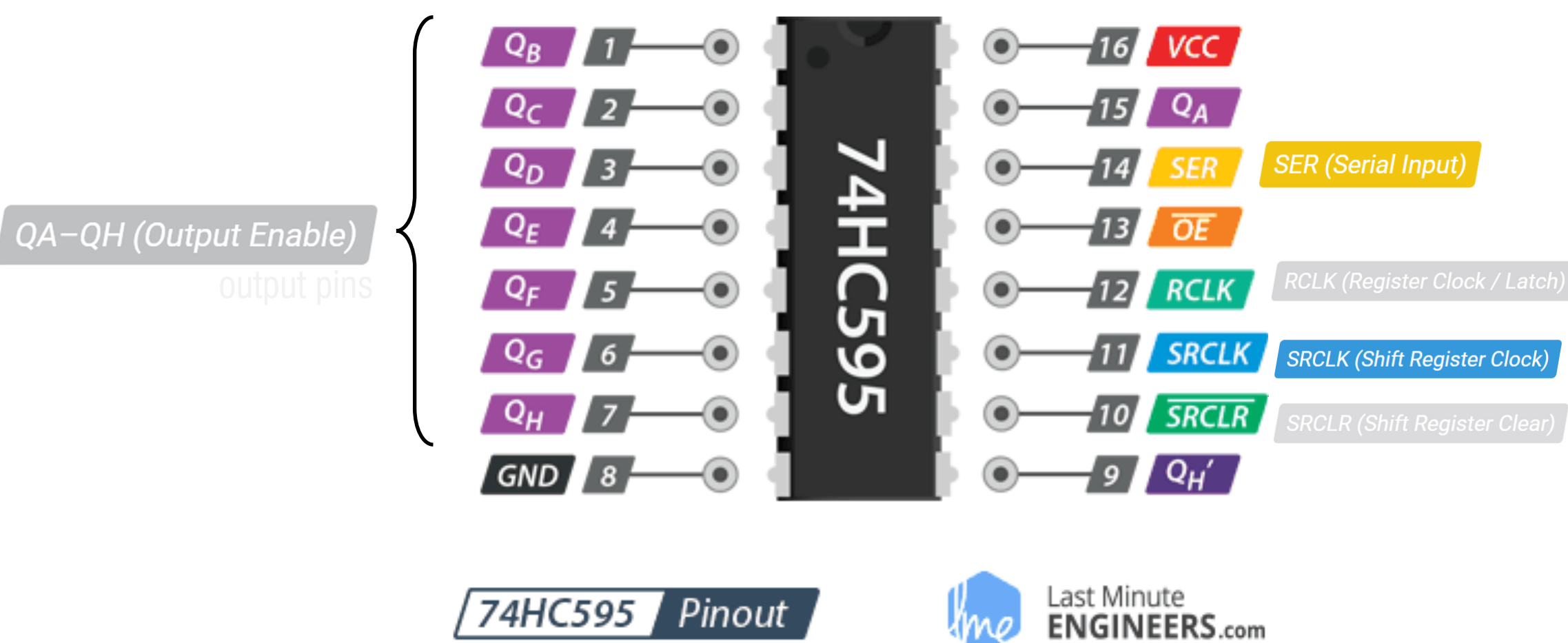
when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register

the clock for the shift register

allows us to reset the entire Shift Register to 0

74HC595 { ^{8bit} Shift Register
^{8bit} Storage Register 0b00001000 Clock ticks 8 times

Shift Register - How – 74HC595



to feed data into the shift register a bit at a time (0)

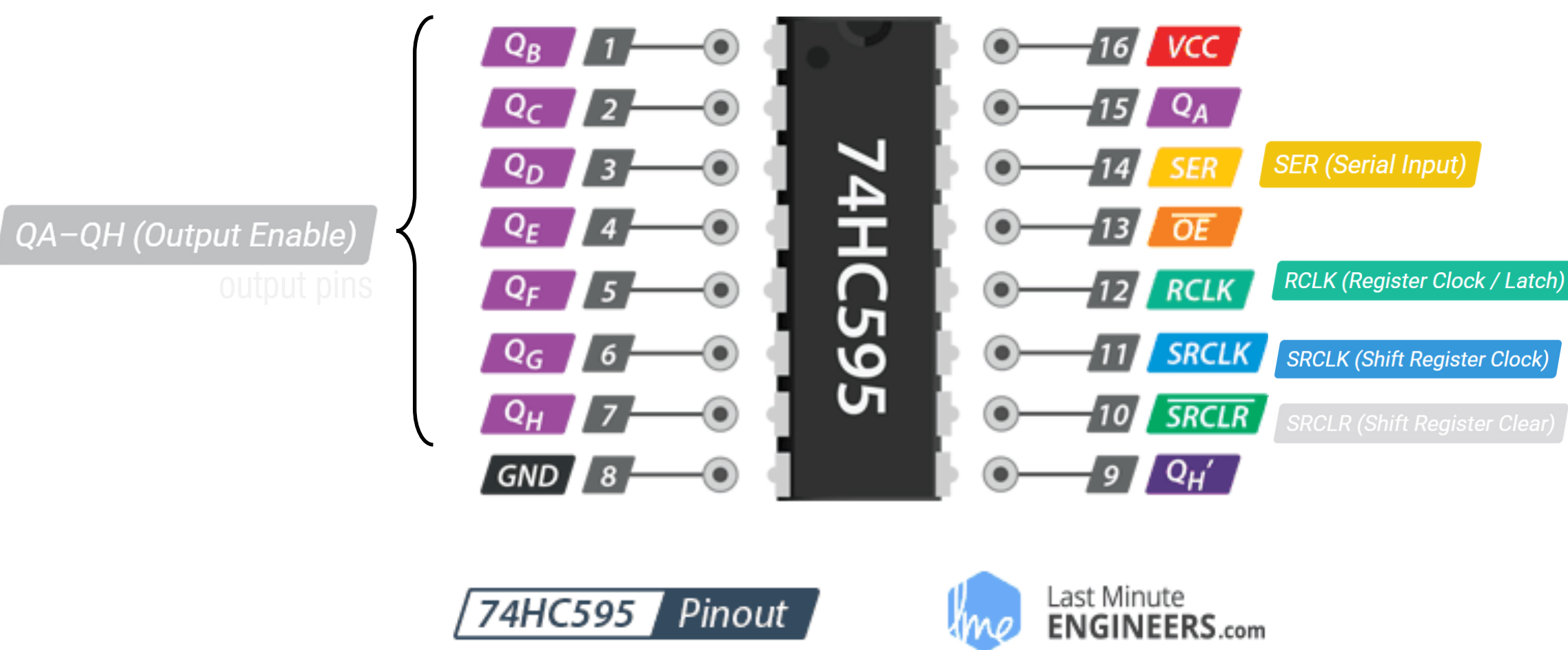
when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register

the clock for the shift register

allows us to reset the entire Shift Register to 0



Shift Register - How – 74HC595



to feed data into the shift register a bit at a time

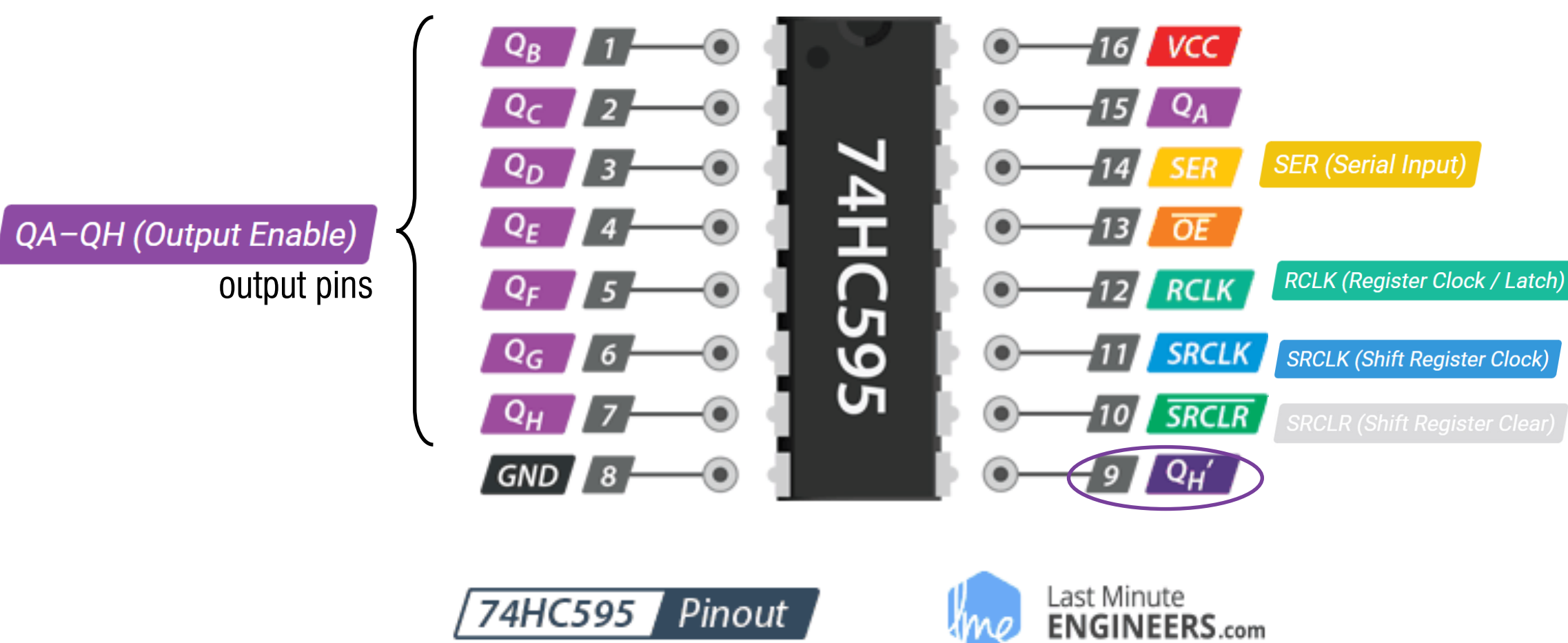
when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register

the clock for the shift register

allows us to reset the entire Shift Register to 0



Shift Register - How – 74HC595



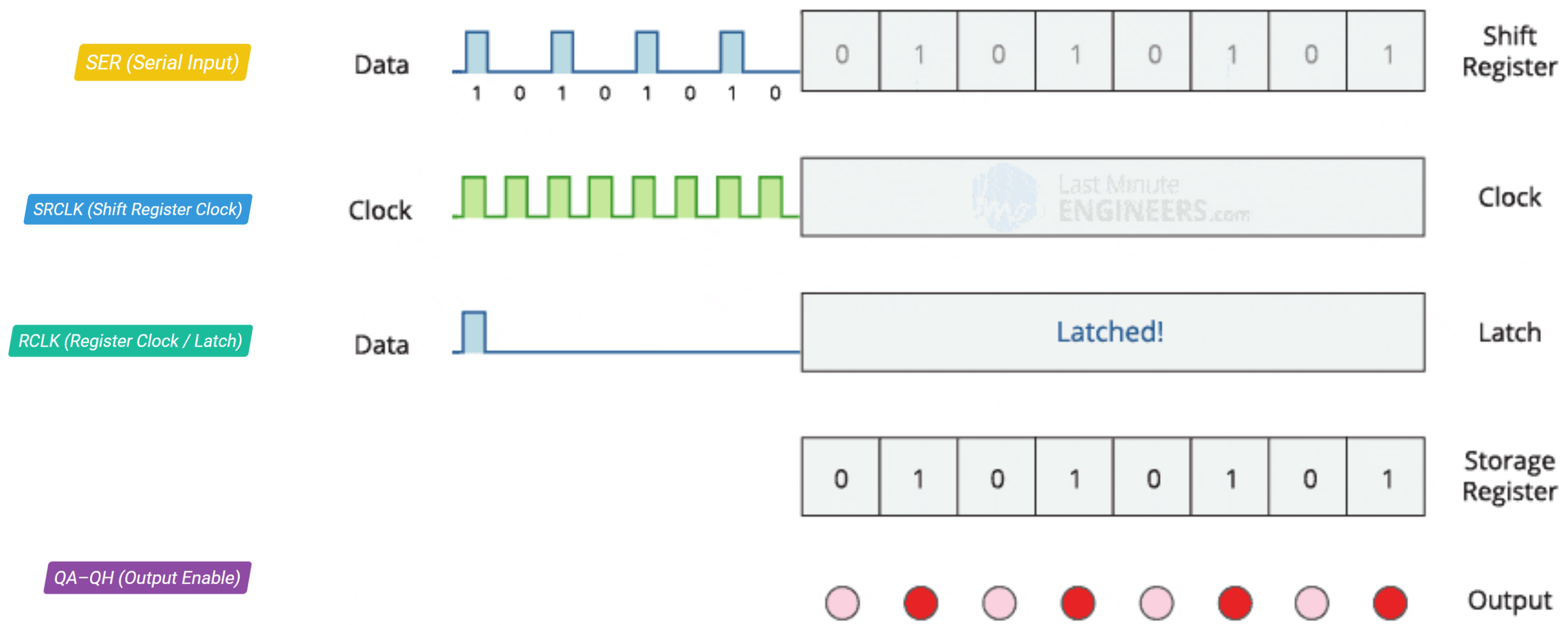
to feed data into the shift register a bit at a time

when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register
the clock for the shift register

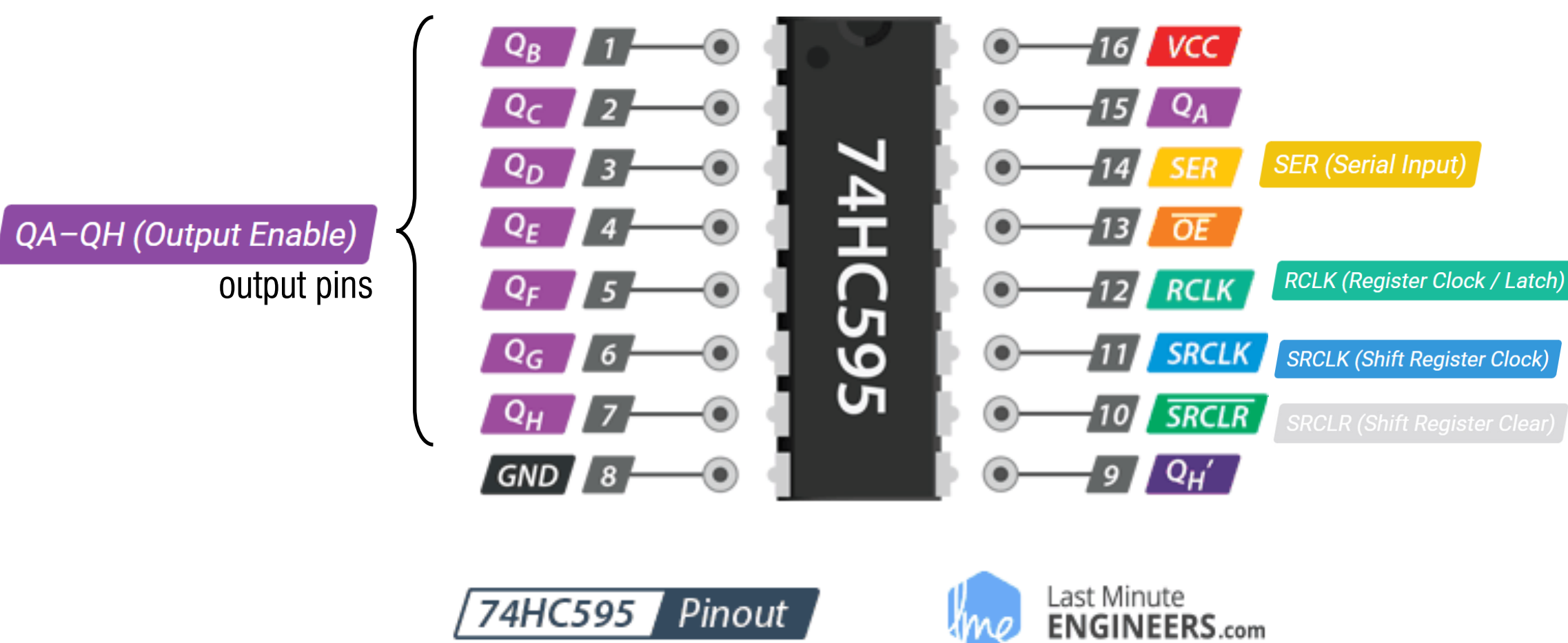
allows us to reset the entire Shift Register to 0



Shift Register - How – 74HC595



Shift Register - How – 74HC595



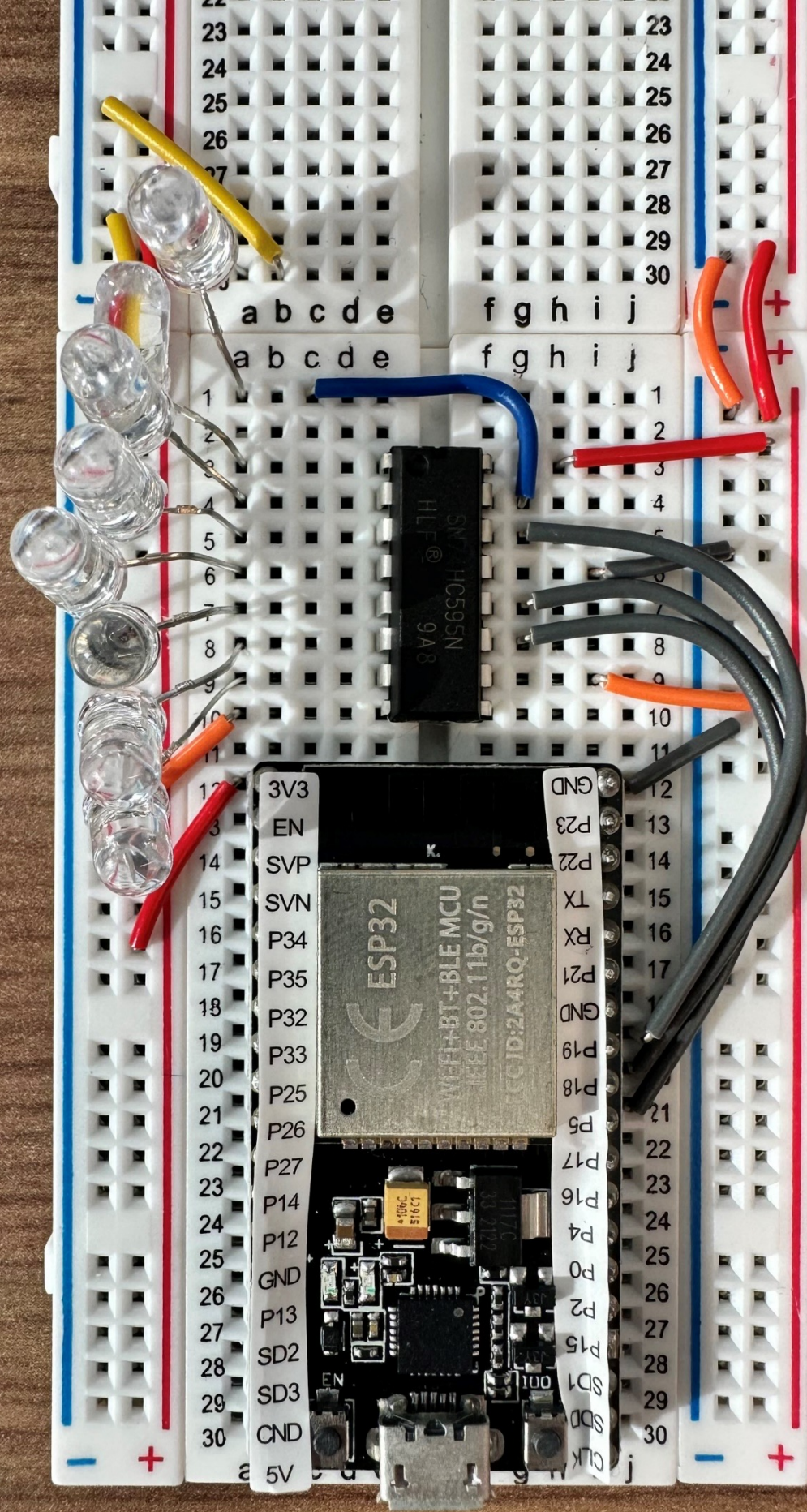
to feed data into the shift register a bit at a time

when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register
the clock for the shift register

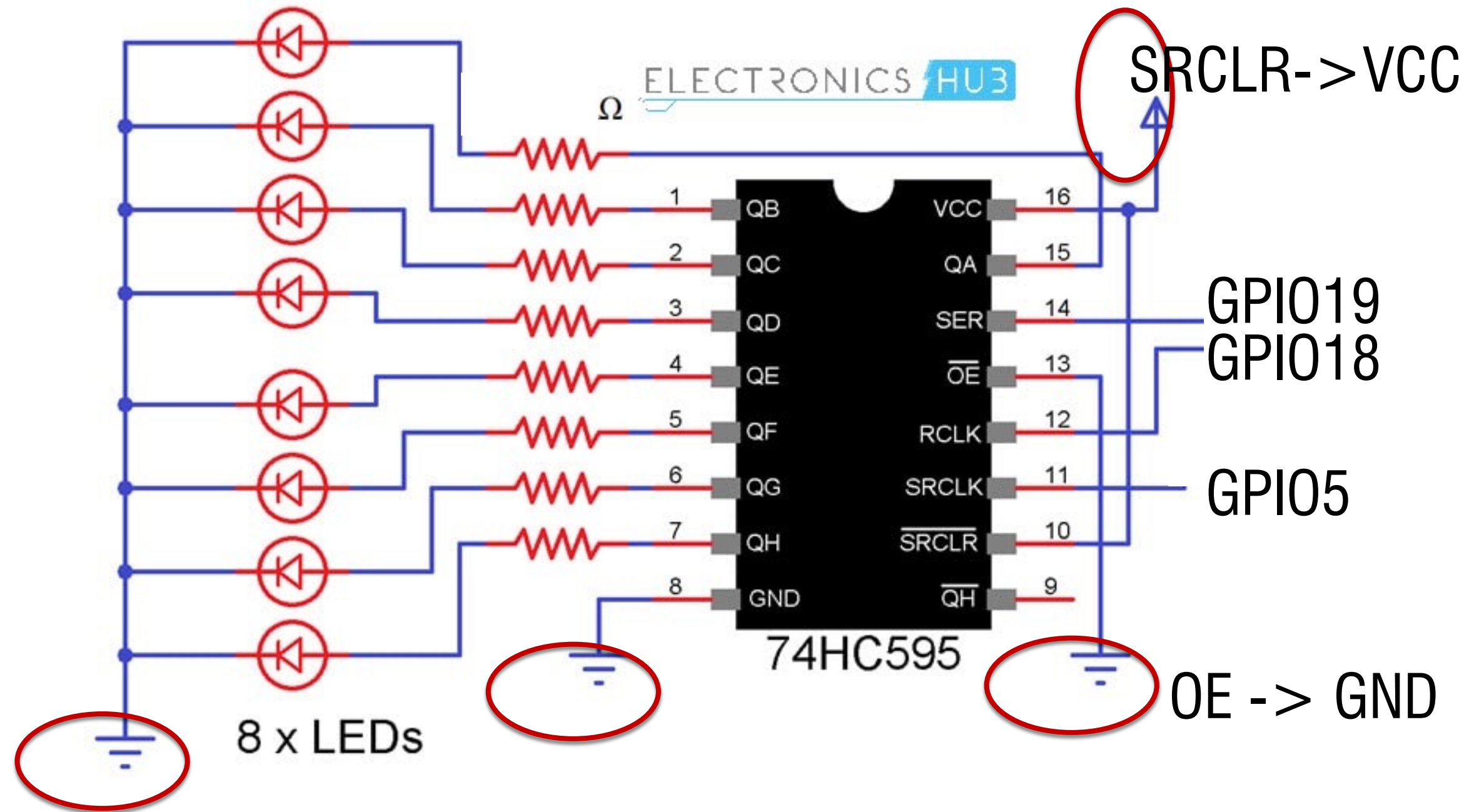
allows us to reset the entire Shift Register to 0



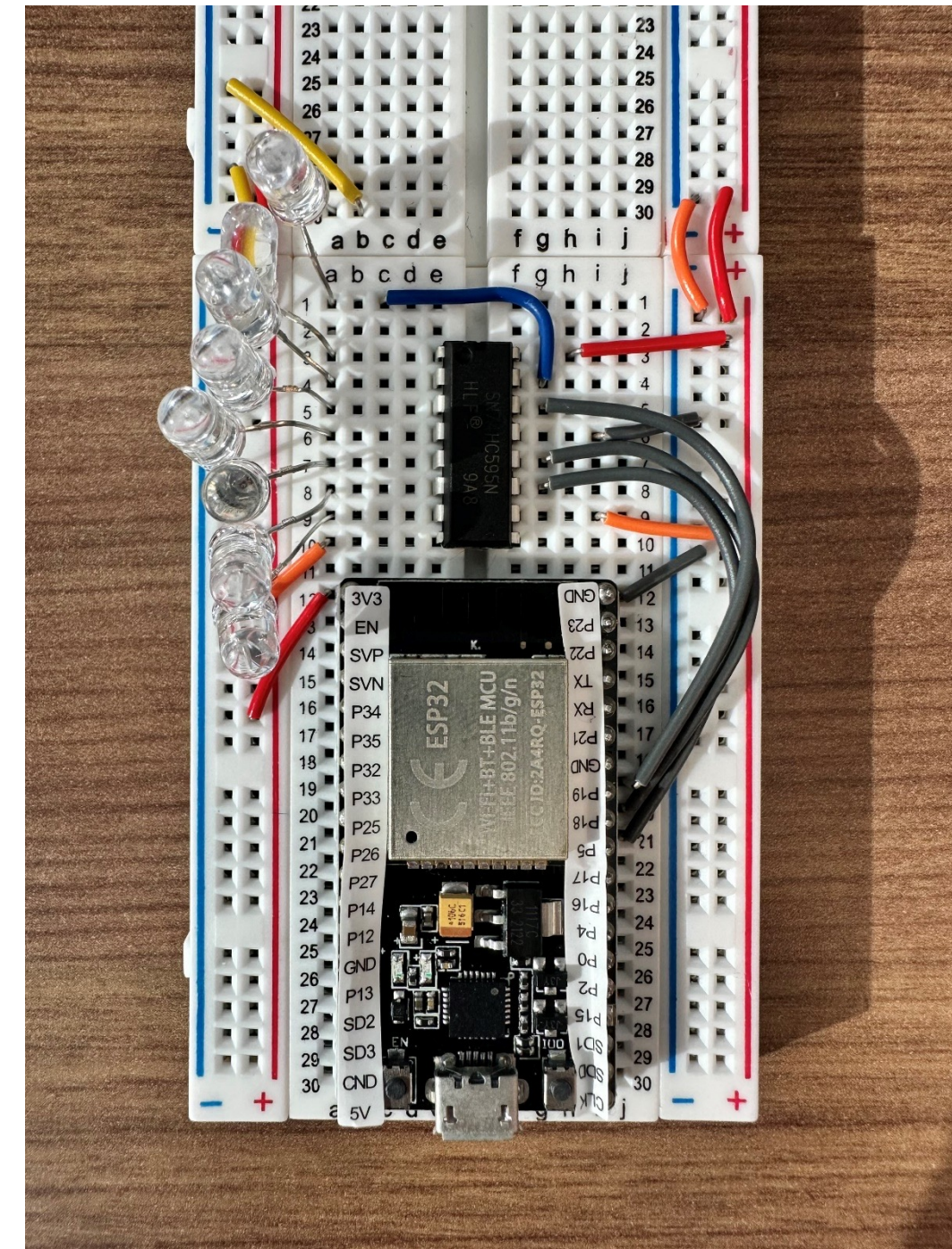
74HC595 - Wiring



74HC595 - Wiring



SER -> Data
RCLK -> Latch
SRCLK -> Clock



74HC595 - Basic coding - light up one LED

```
int latchPin = 18; // Latch pin of 74HC595
int clockPin = 5; // Clock pin of 74HC595
int dataPin = 19; // Data pin of 74HC595
```

```
void setup()
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
}
```

```
void loop()
{
  digitalWrite(dataPin, HIGH);
  digitalWrite(clockPin, HIGH);
  digitalWrite(clockPin, LOW);
```

```
  for(int i=0; i<7; i++)
  {
    digitalWrite(dataPin, LOW);
    digitalWrite(clockPin, HIGH);
    digitalWrite(clockPin, LOW);
  }
```

```
  digitalWrite(latchPin, HIGH);
  digitalWrite(latchPin, LOW);
```

```
  delay(4000);
}
```

0b10000000

}

Push one bit of '1' into the shift register
Tick the clock pin

}

Shift seven bit of '0' into the shift register

Tick the clock pin each time a new data is in

}

Set the latch pin to push data to the storage register that sent to output

bitSet()

[Bits and Bytes]

Description

Sets (writes a 1 to) a bit of a numeric variable.

Syntax

```
bitSet(x, n)
```

Parameters

`x`: the numeric variable whose bit to set.

`n`: which bit to set, starting at 0 for the least-significant (rightmost) bit.

Returns

`x`: the value of the numeric variable after the bit at position `n` is set.

```
byte myByte = 0b00001100; // Initial value: 12 in binary
```

```
bitSet(myByte, 2);
```

```
void setup() {  
    Serial.begin(9600);  
    Serial.println(myByte, BIN);  
}
```

74HC595 - Controlling shift register with built-in functions

```
byte led_data = 0;           // Variable to hold the pattern of which LEDs are currently turned on or off

void setup()
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
}

void loop()
{
  led_data = 0;
  updateShiftRegister();
  delay(500);

  for(int i=0; i<8; i++)
  {
    bitSet(led_data, i);      // Set the bit that controls that LED in the variable 'led_data'
    updateShiftRegister();
    delay(500);
  }
}

void updateShiftRegister()
{
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, LSBFIRST, led_data);
  digitalWrite(latchPin, HIGH);
}
```

//Shifts out a byte of data one bit at a time
//putting the latch Pin HIGH

Assignment – Nov 15 EOD

