On Monday

We will do a laser cutter training (on different ways of using it). Later we will post the training details on Piazza.
Shift Register

Huaishu Peng | UMD CS | Fall 2023
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.

https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/
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

Output pins 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.
3. When driven HIGH, the contents of the Shift Register are copied into the Storage/Latch Register.
4. The clock for the shift register allows us to reset the entire Shift Register to 0.
5. The output pins of the Shift Register can be used to feed data into the shift register bit by bit.

### 74HC595 Pinout

- **Q_A** (Osr): Output register enable
- **Q_B** to **Q_H**: 8 output pins
- **SER**: Serial input
- **OE**: Output enable
- **RCLK**: Register clock/latch
- **SRCLK**: Shift register clock
- **SRCLR**: Shift register clear

### Example

To set up the Shift Register with a binary value of 0b00000001:

1. Connect the **OE** pin to 5V.
2. Connect the **SER** pin to a 5V source.
3. Apply the binary value of 0b00000001 to the output pins.
4. Apply the clock signal to the RCLK pin.

### Conclusion

The 74HC595 is a convenient component for managing data in serial form, enabling efficient data transfer in digital systems.
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 Pinout

74HC595

8bit Shift Register

8bit Storage Register

0b00000010

https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/
Shift Register - How – 74HC595

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

The clock for the shift register allows us to reset the entire Shift Register to 0 when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register.

8-bit 8-bit Shift Register Storage Register 0b00000100

https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/
Shift Register - How – 74HC595

- Output pins to feed data into the shift register a bit at a time (0)
- The clock for the shift register allows us to reset the entire Shift Register to 0 when driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register.

8-bit Shift Register
8-bit Storage Register
0b00001000 Clock ticks 8 times

https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/
Shift Register - How – 74HC595

Output pins 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
Storage Register
0b10000000

https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/
Shift Register - How – 74HC595

Output pins to feed data into the shift register 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 Pinout

74HC595 8bit Shift Register  0b10000000
74HC595 8bit Storage Register  0b10000000

https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/
Shift Register - How – 74HC595

When driven HIGH, the contents of Shift Register are copied into the Storage/Latch Register, allowing us to reset the entire Shift Register to 0.

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

Shift Register

Storage Register 74HC595

0b10000000

8bit

Shift Register

0b10000000

8bit

Storage Register

https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/
Shift Register - How – 74HC595

SER (Serial Input)

SRCLK (Shift Register Clock)

RCLK (Register Clock / Latch)

QA-QH (Output Enable)

https://lastminuteengineers.com/74hc595-shift-register-arduino-tutorial/
74HC595 - Wiring

- SRCLR -> VCC
- GPIO19
- GPIO18
- GOIO5
- OE -> GND
- SER -> Data
- RCLK -> Latch
- SRCLK -> Clock
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);
}

{ 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
}
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); //Shifts out a byte of data one bit at a time
    digitalWrite(latchPin, HIGH); //putting the latch Pin HIGH
}