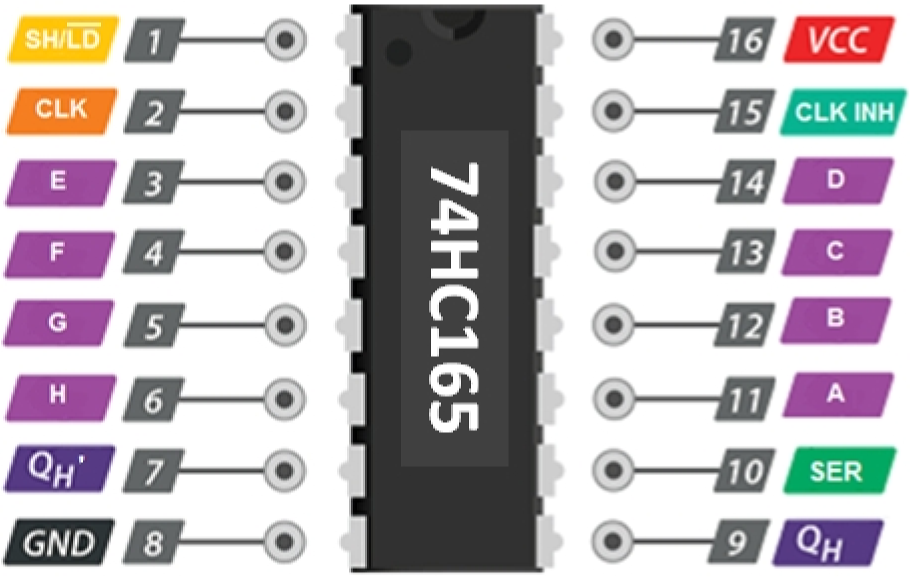
**STEMKRAF – 74HC165 ( Shift Register with 8-bit INPUT )**
https://github.com/teaksoon/stemkraf

Each 74HC165 Shift Register IC provides 8 digital input for a single micro-controller digital input pin. Each 74HC165 chip can be “chained” to another 74HC165 chip. That means we can have multiples of 8 additional digital input pins from each “chained” 74HC165chip. 1chip=8input, 2chips=16input, 3chips=24input and so on...



**GND** – To Arduino GND
**VCC** – To Arduino VCC
**A to H** – Each chip pin is connected to one Input Device. If not used with any Input Device, connect to GND on Arduino.

**SH/LD** – Connect to any Arduino Pin
Set this Pin to HIGH for chip to go into “Shift Mode”
Set this Pin to LOW for chip to go into “Load Mode”

**CLK INH** – Connect to any Arduino Pin
This is used with SH/LD “Load Mode”. When CLK INH change from HIGH to LOW, it will load Input Device digital value into the respective chip A to H pins

**CLK** – Connect to any Arduino Pin
This is used with SH/LD “Shift Mode”. When CLK change from HIGH to LOW, shifting happens. Value in chip pin H goes into QH, G into H, F into G,... A into B.

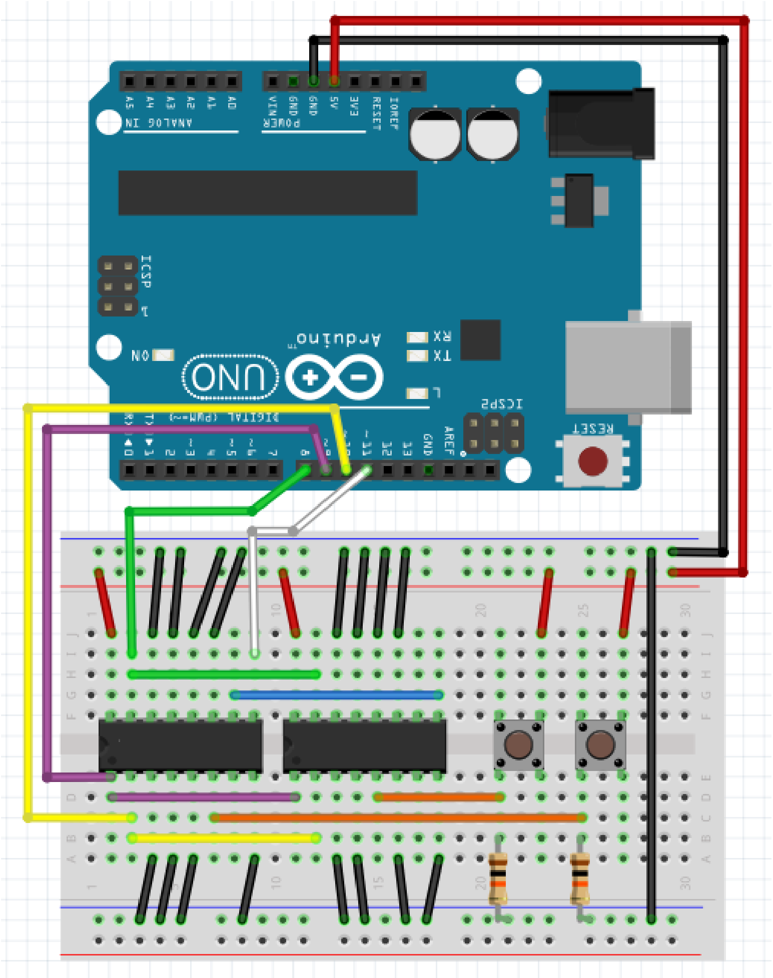
**QH** – Connect to any Arduino Pin
Gets the ‘H’ Pin value after shifting. Only the first chip in multiple chained 74HC165 is connected to Arduino Pin. Second chip owwards are connected from previous chip **SER** pin

**QH’** – Normally this is not used
It stores the opposite of QH, if QH=0, then QH’=1, if QH=1 then QH’=0

**SER** – This is for multiple chips chain use. To be connected to the QH pin of next chained chip

**STEMKRAF - 74HC165 ( Shift Register for 8-bit INPUT )**
https://github.com/teaksoon/stemkraf

```
Program: stemkraf_74HC165
   (1/2): test program for 74HC165
        :
        : by TeakSoon Ding for STEMKRAF (NOV-2021)
```



**Hardware:**
1x Arduino Uno
1x Solderless Breadboard

   Jumper wires

2x tactile switch
2x Resistor 10Kohm

2x 74HC165 Shift Register

Each 74HC165 chip have 8 Input pins. This setup has two 74HC165 chips. This means we will have 16 Input pins for our Digital Input device. In this setup, we have two 2 tactile buttons attached to one pin on each chip for Input. We have 14 unused Input pins.

All the unused 74HC165 Input Pin should be connected to GND, so that we get a constant 0 for those pins, otherwise we get random values which is harmless but can be confusing.

The number Arduino Uno pin used will be the same for 1 chip or multiple chained chips. We will need 4 arduino pin to 1 chip, to get 8 Input Pin, if we have 2 chips, we still use the same 4 arduino pin to get 16 Input Pin, if we use 10 chips we also use the same 4 arduino pin to get 80 Input Pin.

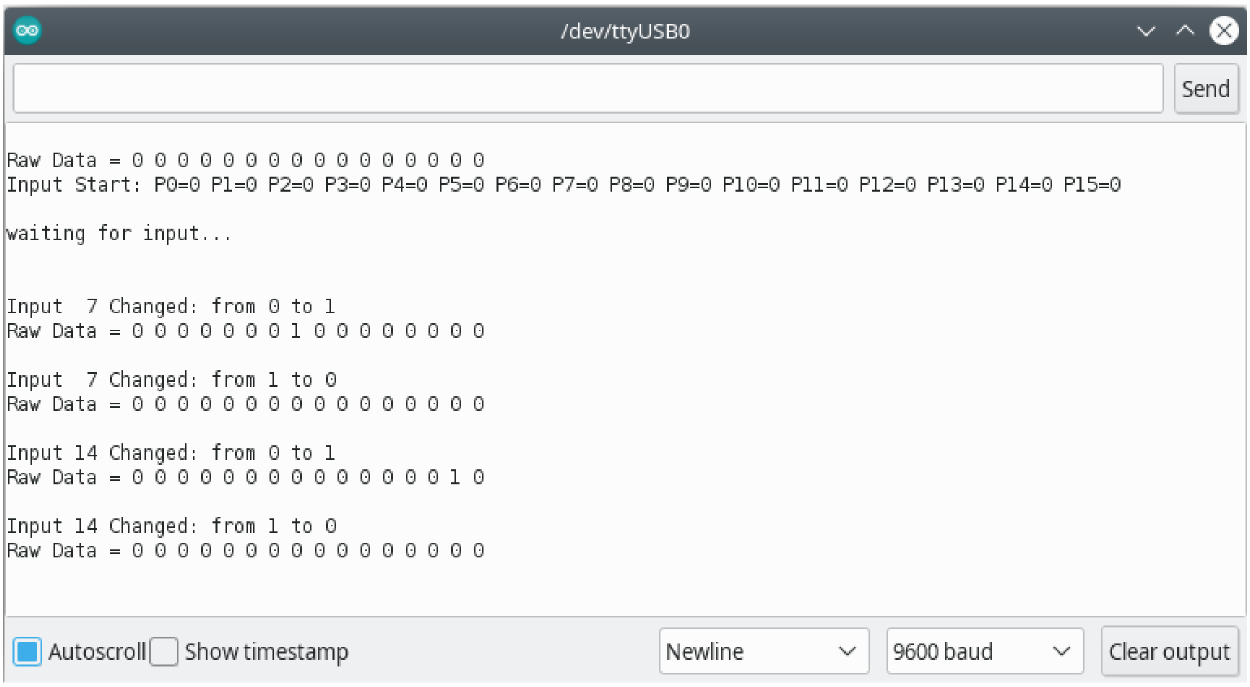
Each chip has Input pin labelled as A to H, you can connect any digital Input device to each of them. Since all chips have pin A to H, when we use multiple chips, it is easier to use an index numbering system. Later can just retrieve by running a function like **sreg_get_state(14)** to get the the input device connected to **Chip 2, pin G**

| | | |
|---|---|---|
| Chip 1: A = 0 | Chip 2: A = 8 | Chip 3: A = 16 |
| Chip 1: B = 1 | Chip 2: B = 9 | Chip 3: B = 17 |
| Chip 1: C = 2 | Chip 2: C = 10 | ... |
| Chip 1: D = 3 | Chip 2: D = 11 | ... |
| Chip 1: E = 4 | Chip 2: E = 12 | ... |
| Chip 1: F = 5 | Chip 2: F = 13 | ... |
| Chip 1: G = 6 | **Chip 2: G = 14** | ... |
| Chip 1: H = 7 | Chip 2: H = 15 | ... |

**STEMKRAF - 74HC165 ( Shift Register for 8-bit INPUT )**
https://github.com/teaksoon/stemkraf

```
Program: stemkraf_74HC165
  (2/2): test program for 74HC165
       :
       : by TeakSoon Ding for STEMKRAF (NOV-2021)
```

- Upload this program with the Arduino IDE Software
- Open up the Serial Monitor from the Arduino IDE Software
- Watch the Serial Monitor Screen while pressing/releasing the tactile button



```
Raw Data = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Input Start: P0=0 P1=0 P2=0 P3=0 P4=0 P5=0 P6=0 P7=0 P8=0 P9=0 P10=0 P11=0 P12=0 P13=0 P14=0 P15=0

waiting for input...


Input  7 Changed: from 0 to 1
Raw Data = 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0

Input  7 Changed: from 1 to 0
Raw Data = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Input 14 Changed: from 0 to 1
Raw Data = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0

Input 14 Changed: from 1 to 0
Raw Data = 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

In this test program, each Output pin is reference by a pin index position ( iPos ), starting from 0 to 7 for the first chip, second chained chip 8 to 15, third chained chip 16 to 23 and so on...

There are 2 main function in this program,

**sreg_74hc165_read()** - This function will read the state of all the chip Input pin and store them in a global 8-bit array name **sreg**

**sreg_get_state(iPos)** - this function will retrieve the state for pin index position, from the **sreg** global array ( **you must run the sreg73hc165_read() function first to load the sreg array before using this function** )

iPos = 0 to ... ( max iPos depends on number of chips connected )

This program uses an 8-bit (one byte) array for working storage, this is to easily manage the chained chips. Each byte (8-bit) in the array is for each Chip (8-Pin).

To have more chips, we only need to change #define TOTAL_CHIP in this program

Some of you may be wondering why I did not have any delaymicroseconds() in my code ( between the HIGH/LOW ). I think they are not necessary. Why ?

The shift register operating at 5V with Arduino, requires less than 30 nano-seconds for state change. The Atmega328 chip with Arduino is running at 16Mhz, which requires 62.5 nano-second to complete one clock-cycle. To set a Pin to HIGH or LOW, we will surely need more than one clock cycle, that is more than enough for the 30 nano-seconds shift register requirement.

Since we are using digitalWrite() function, each digitalWrite() function requires 50 clock-cycles, that is more than 3000 nano-seconds.