

Interface Shift Register with Atmega

74HC595 (8Bits)

Preface

- When there more devices to drive than available MCU pins, this shift register comes handy
- Essentially a serial to parallel converter
- Typical applications include driving 7 Segment display, Bargraph display etc.
- One or more shift registers can be cascaded with the existing one (daisy chaining) to get 16 or 32 or more separate output lines without having to increase MCU pins
- These work as output devices. There are devices available for input as well

Reference

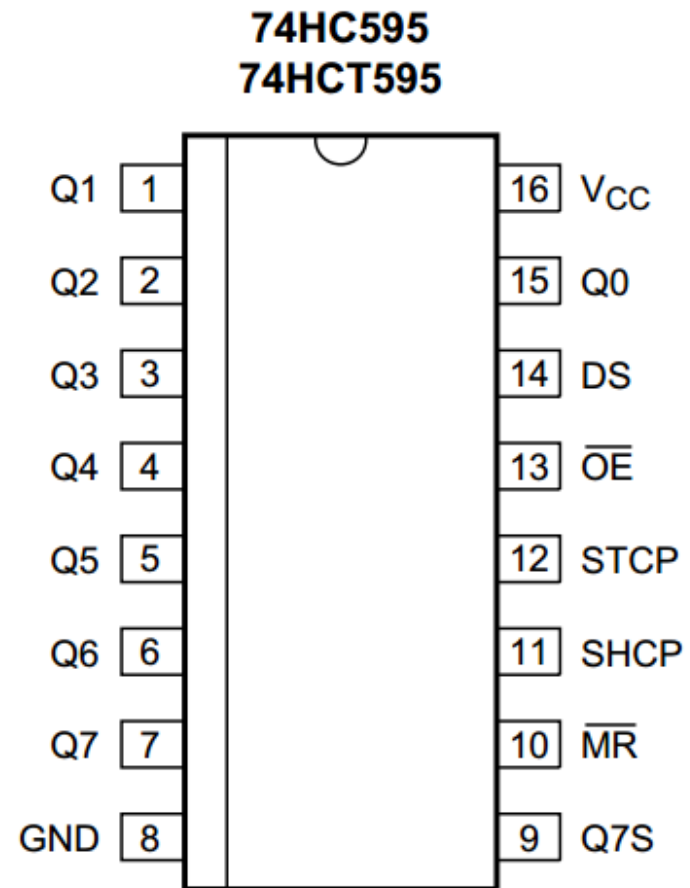
- Datasheet: E:\educational\datasheets\74HC_HCT595.pdf



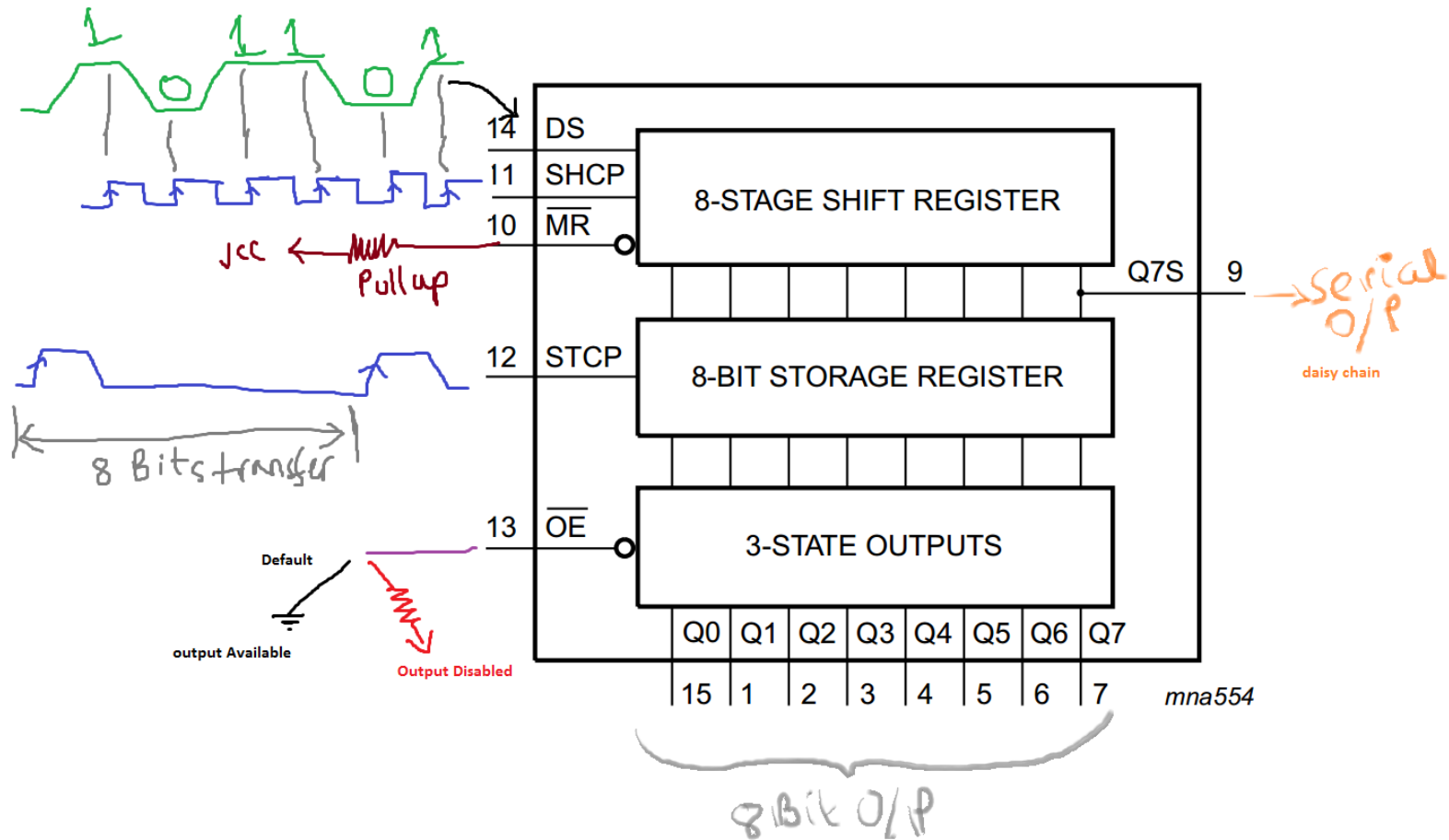
74HC_HCT595.pdf

- Extreme-Electronics: <http://extremeelectronics.co.in/avr-tutorials/using-shift-registers-with-avr-micro-avr-tutorial/>
- Youtube : <https://www.youtube.com/watch?v=d7Au3l4ZdZc>

Pin out



Working mechanism



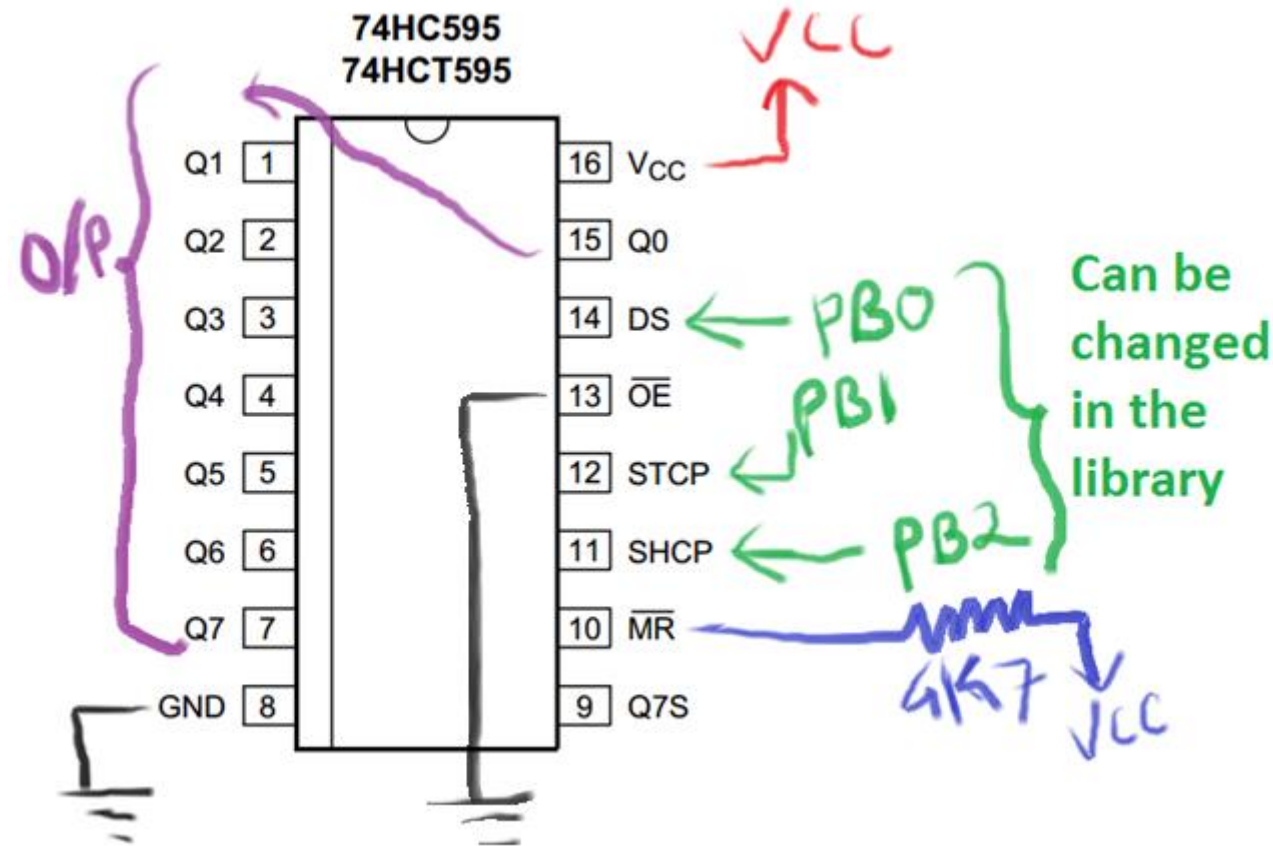
Working Principle DS and SHCP

- DS (serial data pin-14) takes in serial data.
- At every rising edge of shift clk on pin 11, the register left-shifts its previous byte present in 8 STAGE SHIFT REGISTER.
- samples and captures DS (high or low) and places it on newly vacant LSB position
- This way 8 rising edge on SHCP will load a new byte in 8 STAGE SHIFT REGISTER.

Working Principle STCP

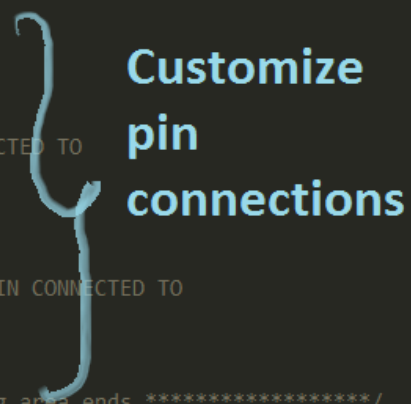
- Unless **STCP** receive a rising edge the content of **8 STAGE SHIFT REGISTER** will not be available at **8 BIT STORAGE REGISTER**
- Once **STCP** receives a rising edge the content of **8 STAGE SHIFT REGISTER** gets copied into **8 BIT STORAGE REGISTER**
- So its usually done once after 8 new bits have shifted in **8 STAGE SHIFT REGISTER**
- When **OE** is connected to GND (default) the 8 bits in **8 BIT STORAGE REGISTER** is available in **3 STATE OUTPUTS** for external use

MCU connection (Basic mode) just 5 pins (3MCU pins)



Library (ShiftRegHC595.h)

```
1
2  /* HC595 Shift register needs Three MCU lines to work and GND, VCC
3
4  DS (Serial data PIN-14) is connected to serial data output pin d
5  SH_CP (shift register clk input PIN-11) data bit on DS is captur
6  ST_CP (Storage Register clk input PIN-12) at rising clk the 8 bi
7
8  VCC (PIN-16)
9  GND (PIN-8)
10 MR` ( Master Reset active low PIN-10). Usually pulled up. if con
11 */
12
13
14 /***** user configureable area *****/
15 // DS PIN CONNECTED TO
16 #define HC595_DS_DDR    DDRB
17 #define HC595_DS_PORT   PORTB
18 #define HC595_DS_POS    PB0
19
20 // SH_CP (Shift Clk) PIN CONNECTED TO
21 #define HC595_SHCP_DDR  DDRB
22 #define HC595_SHCP_PORT PORTB
23 #define HC595_SHCP_POS  PB1
24
25 // ST_CP (Storage/Latch Clk) PIN CONNECTED TO
26 #define HC595_STCP_DDR  DDRB
27 #define HC595_STCP_PORT PORTB
28 #define HC595_STCP_POS  PB2
29 /***** user config area ends *****/
30
31 |
32
```



Customize
pin
connections

Library (ShiftRegHC595.h)

```
36
37 //Initialize HC595 Shift Register
38 void HC595Init() {
42 }
43
44
45 //Low level macros to change data (DS)lines
46 #define HC595DataHigh() (HC595_DS_PORT|=(1<<HC595_DS_POS))
47 #define HC595DataLow() (HC595_DS_PORT&=~(1<<HC595_DS_POS))
48
49
50 //Sends a single clk pulse (rising edge) on SH_CP (shift Clk) line
51 void HC595Pulse() {
54 }
55
56
57 //Sends a single clk pulse (rising edge) on ST_CP (Storage/latch Clk) line
58 void HC595Latch() {
64 }
65
66
67
68 // This is the function that is going to be called by users.
69 void HC595Write(uint8_t data) {
89 }
90
```

Internal Functions

User Function

Library (ShiftRegHC595.h)

```
100 #include <avr/io.h>
101 #include <util/delay.h>
102
103 #include "ShiftRegHC595.h"
104
105
106
107 void main(){
108     //Initialize HC595 system
109     HC595Init();
110     while(1) {
111         HC595Write(0b00110101); //Write the data to HC595
112         _delay_ms(500);           //Wait
113         HC595Write(0b11001010);
114         _delay_ms(500);
115     }
116 }
117
118
119
120
```

Fully running code Example

Extended & Advanced Features

- Sometimes one shift register is not enough.
- So there are four different ways to connect multiple shift registers with MCU
 1. Daisy Chain
 2. Common clocks(SHCP & STCP common and different DS lines)
 3. Common latch (STCP common and different SHCP and DS lines)
 4. Totally separate modules

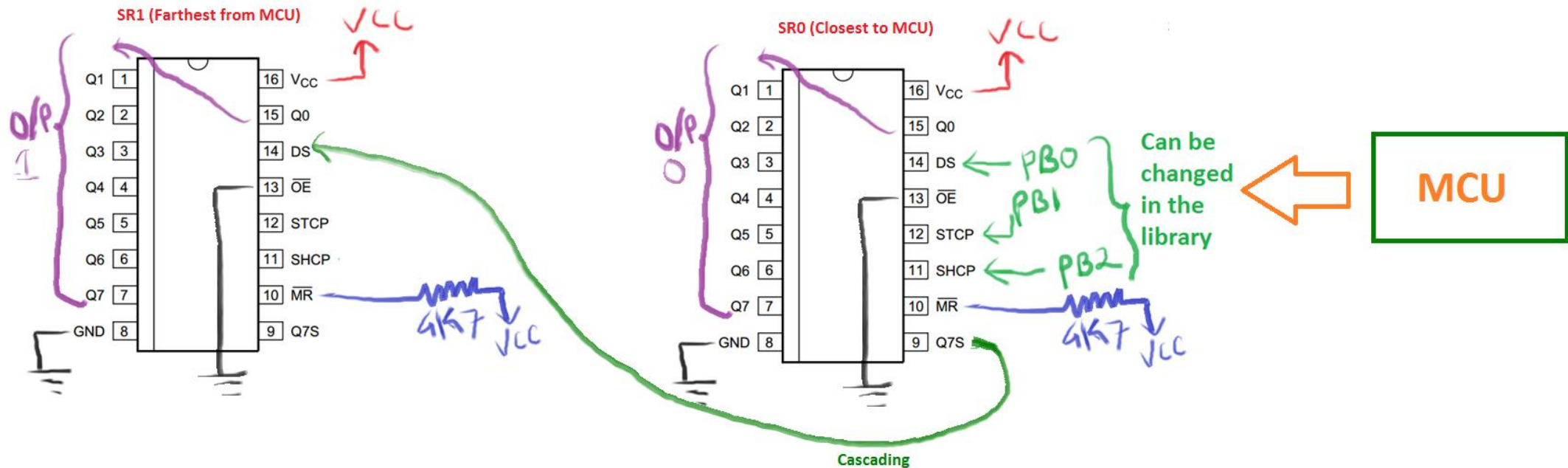
1. Daisy Chain

Cascade

1. Daisy Chain

- Shift registers are cascaded.
- Number of MCU lines doesn't increase(stays 3).
- 2 SR gives 16 o/p lines, 3 gives 24 etc.
- But total number of bits available have to be updated together
- Best suited for display devices (7 segments, bargraphs etc.)

1. Daisy chain Connections



1. Daisy chain Library (shiftrc595_daisy.h)

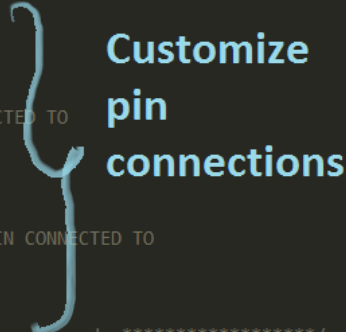
- Include “ShiftRegHC595_daisy.h” and this will automatically include “ShiftRegHC595.h”

```
76
77 #include <avr/io.h>
78 #include <util/delay.h>
79
80 #include "ShiftRegHC595_daisy.h"
81
82
83
```


1. Daisy chain Library (shiftrghc595_daisy.h)

- Then configure the connection for SR0 (Closest to MCU) in [ShiftRegHC595.h](#)

```
1
2  /* HC595 Shift register needs Three MCU lines to work and GND, VCC
3
4     DS (Serial data PIN-14) is connected to serial data output pin 0
5     SH_CP (shift register clk input PIN-11) data bit on DS is captured
6     ST_CP (Storage Register clk input PIN-12) at rising clk the 8 b
7
8     VCC (PIN-16)
9     GND (PIN-8)
10    MR` ( Master Reset active low PIN-10). Usually pulled up. if con
11  */
12
13
14  /***** user configureable area *****/
15  // DS PIN CONNECTED TO
16  #define HC595_DS_DDR    DDRB
17  #define HC595_DS_PORT   PORTB
18  #define HC595_DS_POS    PB0
19
20  // SH_CP (Shift Clk) PIN CONNECTED TO
21  #define HC595_SHCP_DDR  DDRB
22  #define HC595_SHCP_PORT PORTB
23  #define HC595_SHCP_POS PB1
24
25  // ST_CP (Storage/Latch Clk) PIN CONNECTED TO
26  #define HC595_STCP_DDR  DDRB
27  #define HC595_STCP_PORT PORTB
28  #define HC595_STCP_POS PB2
29  /***** user config area ends *****/
30
31
32
```



Customize
pin
connections

1. Daisy chain Library (shiftrghc595_daisy.h)

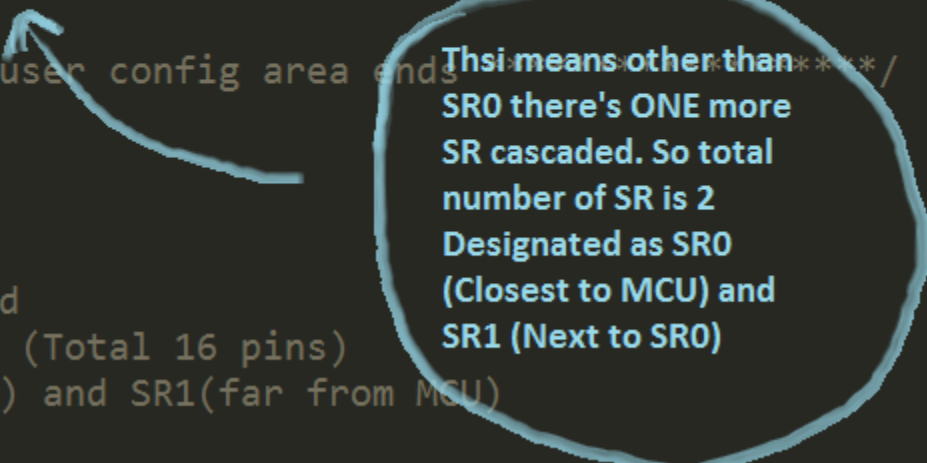
- Then specify number of SR(s) cascaded, i.e. total number of SR(s) excluding SR0, in ShiftRegHC595_daisy.h

```
/* ***** user configureable area ***** */

// if more than one 75HC595 shift registers
// are cascaded then specify number of cascaded shift registers( to
#define DAISY_CHAIN 1 // options 1, 2, 3, (0 => no daisy chain)

/* ***** user config area ends ***** */

// writes double word
// 2 shift registers (Total 16 pins)
// SR0(closer to MCU) and SR1(far from MCU)
#if DAISY_CHAIN == 1
void HC595WriteDW(uint8_t data0, uint8_t data1) {
```



This means other than SR0 there's ONE more SR cascaded. So total number of SR is 2 Designated as SR0 (Closest to MCU) and SR1 (Next to SR0)

1. Daisy chain Library (shiftreghc595_daisy.h)

- That's it, Now just call the functions. Here's a sample program.

```
76
77 #include <avr/io.h>
78 #include <util/delay.h>
79
80 #include "ShiftRegHC595_daisy.h"
81
82
83 void main(){
84     //Initialize HC595 system
85     HC595Init();
86     // first byte goes to SR closest to MCU
87     HC595WriteDW(0b10101010, 0b01010101);
88     while(1) {
89     }
90 }
91
92
```

Hand-drawn diagram illustrating the daisy chain connection of four shift registers (SR0, SR1, SR2, SR3). The output of SR0 (Q7) is connected to the input of SR1. The output of SR1 (Q7) is connected to the input of SR2. The output of SR2 (Q7) is connected to the input of SR3. The output of SR3 is labeled Q0.

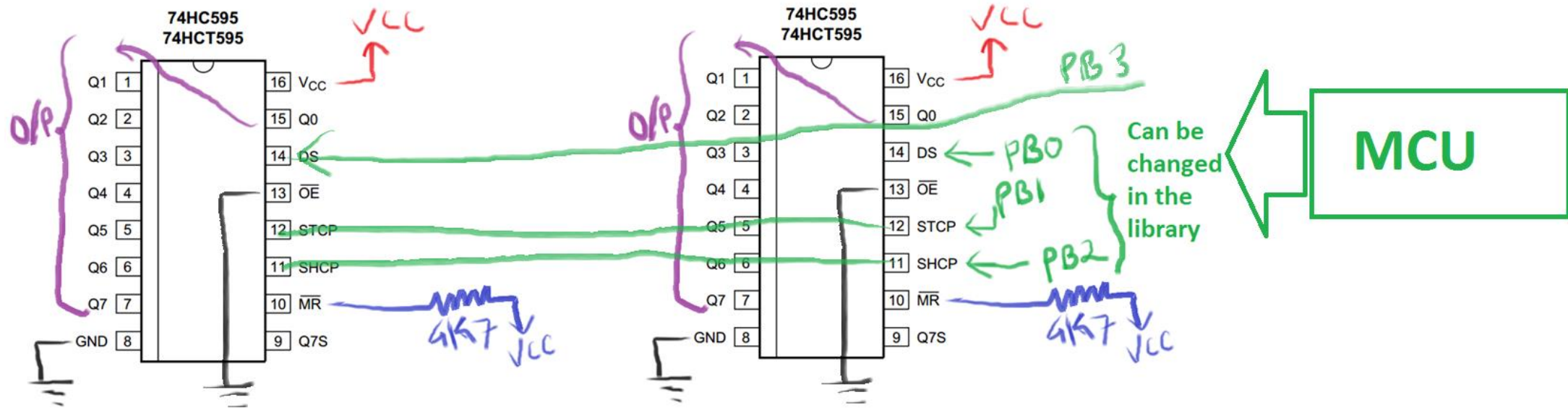
2. Common Clocks

SHCP(shift Clk) & STCP(latch Clk)

2. Common Clocks (SHCP & STCP)

- Shift registers are not cascaded.
- All the SR(s) share the same clock line (SHCP & STCP)
- But each SR have their own an different DS lines
- Since they share clocks. If one SR is updated then this will cause shifts in all the SR(s). To overcome this all shift registers will have to be updated all at once.
- First SR (SR0) requires 3 MCU lines and then after one more MCU lines (DS) each for each SR(s) added.

2. Common Clocks Connections



2. Common Clocks Library (shiftreghc595_comClk.h)

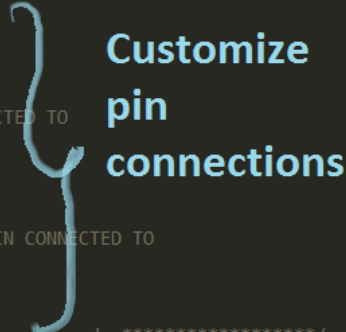
- Include “ShiftRegHC595_comClk.h” and this will automatically include “ShiftRegHC595.h”

```
199 #include <avr/io.h>
200 #include <util/delay.h>
201
202 #include "ShiftRegHC595_comClk.h"
203
204
```

2. Common Clocks Library (shiftrghc595_comClk.h)

- Then configure the connection for SR0 (Closest to MCU) in [ShiftRegHC595.h](#)

```
1
2  /* HC595 Shift register needs Three MCU lines to work and GND, VCC
3
4     DS (Serial data PIN-14) is connected to serial data output pin 0
5     SH_CP (shift register clk input PIN-11) data bit on DS is captured
6     ST_CP (Storage Register clk input PIN-12) at rising clk the 8 bits
7
8     VCC (PIN-16)
9     GND (PIN-8)
10    MR` ( Master Reset active low PIN-10). Usually pulled up. if con
11 */
12
13
14  /***** user configureable area *****/
15  // DS PIN CONNECTED TO
16  #define HC595_DS_DDR    DDRB
17  #define HC595_DS_PORT   PORTB
18  #define HC595_DS_POS    PB0
19
20  // SH_CP (Shift Clk) PIN CONNECTED TO
21  #define HC595_SHCP_DDR  DDRB
22  #define HC595_SHCP_PORT PORTB
23  #define HC595_SHCP_POS PB1
24
25  // ST_CP (Storage/Latch Clk) PIN CONNECTED TO
26  #define HC595_STCP_DDR  DDRB
27  #define HC595_STCP_PORT PORTB
28  #define HC595_STCP_POS PB2
29  /***** user config area ends *****/
30
31
32
```



Customize
pin
connections

2. Common Clocks Library (shiftreghc595_comClk.h)

- Then configure DSx line connections for SR1 and SR2 in [ShiftRegHC595_comClk.h](#)

```
24 // If more than one shift registers are connected (NOT CASCADED)
25 // They can and will share same "shift clk" and "latch clock" lines
26 // But serial data lines will differ. i.e. DS1, DS2, DS3 etc.
27 // In such case specify the additional data lines here.
28
29
30 // DS1 PIN CONNECTED TO
31 #define HC595_DS1_DDR   DDRB
32 #define HC595_DS1_PORT  PORTB
33 #define HC595_DS1_POS   PB3
34
35
36 // // DS2 PIN CONNECTED TO
37 // #define HC595_DS2_DDR   DDRB
38 // #define HC595_DS2_PORT  PORTB
39 // #define HC595_DS2_POS   PB4
40
41
42
43 // ... you can go on and continue to add more shift registers if needed
44
45 /***** user config area ends *****/
46
```

DS1 connection for SR1

If another SR's needed then uncomment and configure

2. Common Clocks Library (shiftrghc595_comClk.h)

- That's it, Now just call the functions. Here's a sample program.

```
196
197
198
199 #include <avr/io.h>
200 #include <util/delay.h>
201
202 #include "ShiftRegHC595_comClk.h"
203
204                                     CK => Common Clock
205
206 void main(){
207     //Initialize HC595 system
208     HC595Init();
209     // first byte goes to SR0 with DS0 line
210     HC595WriteCK(0b10101010, 0b01010101);
211     while(1) {
212     }
213 }
214
```

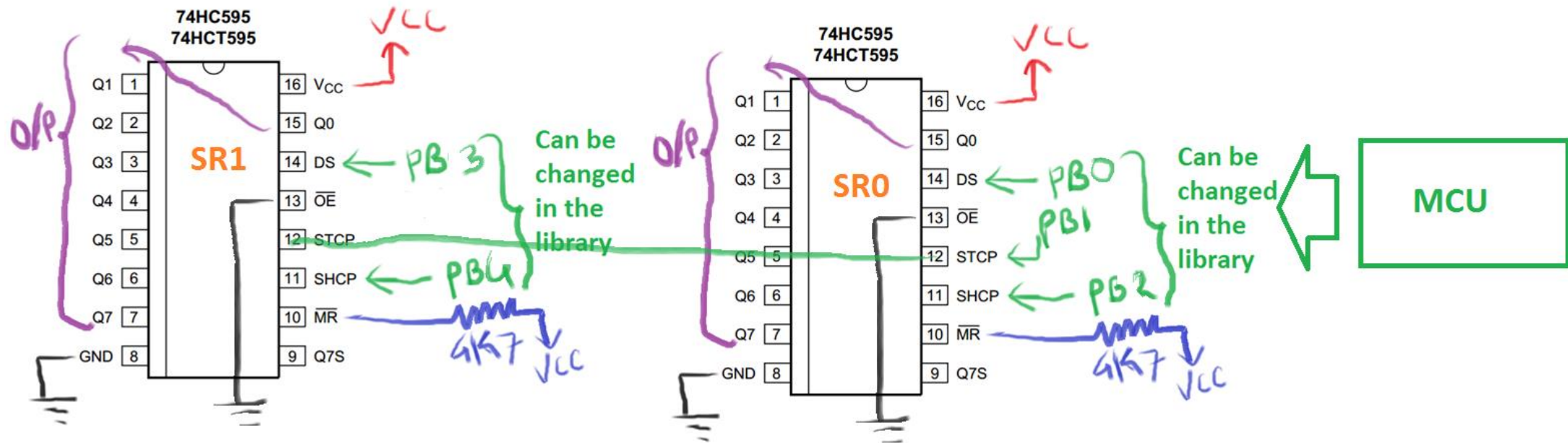
3. Common latch

STCP(latch Clk is common)

3. Common Latch

- The latch Clk(STCP) is shared between all the SR(s)
- Each SR has its own Shift Clk(SHCP) and Serial data(DS)
- This mode can be achieved by Configuring the same PORT, DDR, POS for STCP pic in Totally separate mode

3. Common Latch Connections



3. Common Latch Library (ShiftRegHC595_X.h)

- Include “ShiftRegHC595_X.h” and this will automatically include “ShiftRegHC595.h”

```
180
181 #include <avr/io.h>
182 #include <util/delay.h>
183
184 #include "ShiftRegHC595_X.h"
185
186
187
```

3. Common Latch Library (ShiftRegHC595_X.h)

- Then configure the connection for SR0 (Closest to MCU) in [ShiftRegHC595.h](#)

```
1
2 /* HC595 Shift register needs Three MCU lines to work and GND, VCC
3
4 DS (Serial data PIN-14) is connected to serial data output pin
5 SH_CP (shift register clk input PIN-11) data bit on DS is captu
6 ST_CP (Storage Register clk input PIN-12) at rising clk the 8 b
7
8 VCC (PIN-16)
9 GND (PIN-8)
10 MR* ( Master Reset active low PIN-10). Usually pulled up. if co
11 */
12
13
14 /***** user configureable area *****/
15 // DS PIN CONNECTED TO
16 #define HC595_DS_DDR   DDRB
17 #define HC595_DS_PORT  PORTB
18 #define HC595_DS_POS   PB0
19
20 // SH_CP (Shift Clk) PIN CONNECTED TO
21 #define HC595_SHCP_DDR  DDRB
22 #define HC595_SHCP_PORT PORTB
23 #define HC595_SHCP_POS  PB1
24
25 // ST_CP (Storage/Latch Clk) PIN CONNECTED TO
26 #define HC595_STCP_DDR  DDRB
27 #define HC595_STCP_PORT PORTB
28 #define HC595_STCP_POS  PB2
29 /***** user config area ends *****/
30
31 |
32
```

Customize
pin
connections

PB2

3. Common Latch Library (ShiftRegHC595_X.h)

- Then configure the connection for SR1 in [ShiftRegHC595_X.h](#)

```
26
27 /***** user configureable area *****/
28
29 // 2nd SR-----
30 //DS (Serial data) PIN-14 CONNECTED TO
31 #define HC595_DS1_DDR    DDRB
32 #define HC595_DS1_PORT   PORTB
33 #define HC595_DS1_POS    PB3
34
35 // SH_CP (Shift Clk) PIN-11 CONNECTED TO
36 #define HC595_SHCP1_DDR   DDRB
37 #define HC595_SHCP1_PORT  PORTB
38 #define HC595_SHCP1_POS   PB4
39
40 // ST_CP (Storage/Latch Clk) PIN-12 CONNECTED TO
41 #define HC595_STCP1_DDR   DDRB
42 #define HC595_STCP1_PORT  PORTB
43 #define HC595_STCP1_POS   PB2
44
```

Same PB2

3. Common Latch Library (ShiftRegHC595_X.h)

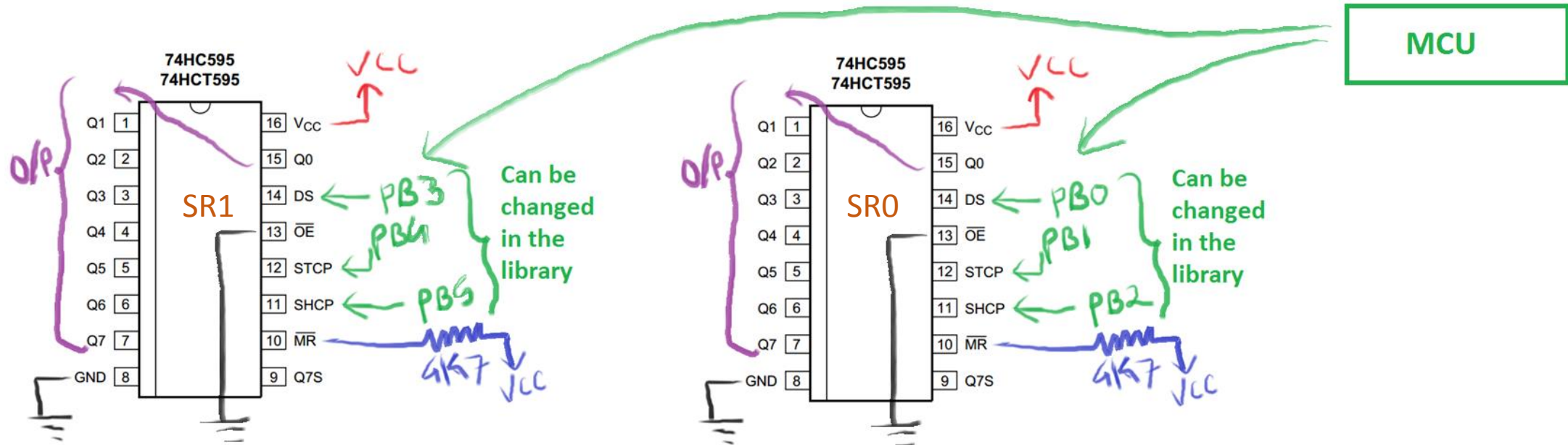
- Sample Program

```
179
180
181 #include <avr/io.h>
182 #include <util/delay.h>
183
184 #include "ShiftRegHC595_X.h"
185
186
187
188 void main(){
189     //Initialize HC595 system
190     HC595Init();
191     HC595Init1();
192     HC595Write(0b10101010);
193     HC595Write1(0b11001100);
194     while(1) {
195     }
196 }
197
198
199
200
```

SRO
SRI

4. Totally Separate

4. Totally separate Connections



4. Totally separate Connections

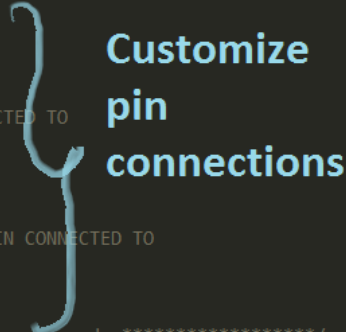
- Include “ShiftRegHC595_X.h” and this will automatically include “ShiftRegHC595.h”

```
180
181 #include <avr/io.h>
182 #include <util/delay.h>
183
184 #include "ShiftRegHC595_X.h"
185
186
187
```

4. Totally separate Connections

- Then configure the connection for SR0 (Closest to MCU) in [ShiftRegHC595.h](#)

```
1
2  /* HC595 Shift register needs Three MCU lines to work and GND, VCC
3
4     DS (Serial data PIN-14) is connected to serial data output pin 0
5     SH_CP (shift register clk input PIN-11) data bit on DS is captured
6     ST_CP (Storage Register clk input PIN-12) at rising clk the 8 bits
7
8     VCC (PIN-16)
9     GND (PIN-8)
10    MR` ( Master Reset active low PIN-10). Usually pulled up. if con
11 */
12
13
14  /***** user configureable area *****/
15  // DS PIN CONNECTED TO
16  #define HC595_DS_DDR    DDRB
17  #define HC595_DS_PORT    PORTB
18  #define HC595_DS_POS    PB0
19
20  // SH_CP (Shift Clk) PIN CONNECTED TO
21  #define HC595_SHCP_DDR  DDRB
22  #define HC595_SHCP_PORT PORTB
23  #define HC595_SHCP_POS  PB1
24
25  // ST_CP (Storage/Latch Clk) PIN CONNECTED TO
26  #define HC595_STCP_DDR  DDRB
27  #define HC595_STCP_PORT PORTB
28  #define HC595_STCP_POS  PB2
29  /***** user config area ends *****/
30
31
32
```




Customize
pin
connections

4. Totally separate Connections

- Then configure the connection for SR1 in [ShiftRegHC595_X.h](#)

```
26
27 /***** user configurable area *****/
28
29 // 2nd SR-----
30 //DS (Serial data) PIN-14 CONNECTED TO
31 #define HC595_DS1_DDR    DDRB
32 #define HC595_DS1_PORT  PORTB
33 #define HC595_DS1_POS    PB3
34
35 // SH_CP (Shift Clk) PIN-11 CONNECTED TO
36 #define HC595_SHCP1_DDR  DDRB
37 #define HC595_SHCP1_PORT PORTB
38 #define HC595_SHCP1_POS  PB4
39
40 // ST_CP (Storage/Latch Clk) PIN-12 CONNECTED TO
41 #define HC595_STCP1_DDR  DDRB
42 #define HC595_STCP1_PORT PORTB
43 #define HC595_STCP1_POS  PB5
44
```

All New pins for SR1

A diagram consisting of three blue arrows. The first arrow starts at the line '#define HC595_DS1_POS PB3' and points towards the text 'All New pins for SR1'. The second arrow starts at the line '#define HC595_SHCP1_POS PB4' and points towards the same text. The third arrow starts at the line '#define HC595_STCP1_POS PB5' and points towards the same text.

4. Totally separate Connections

- Sample Program

```
179
180
181 #include <avr/io.h>
182 #include <util/delay.h>
183
184 #include "ShiftRegHC595_X.h"
185
186
187
188 void main(){
189     //Initialize HC595 system
190     HC595Init();
191     HC595Init1();
192     HC595Write(0b10101010);
193     HC595Write1(0b11001100);
194     while(1) {
195     }
196 }
197
198
199
200
```

→ SRO
→ SRI