

Button Matrix - Documentation

1 Circuit Board

1.1 Schematic

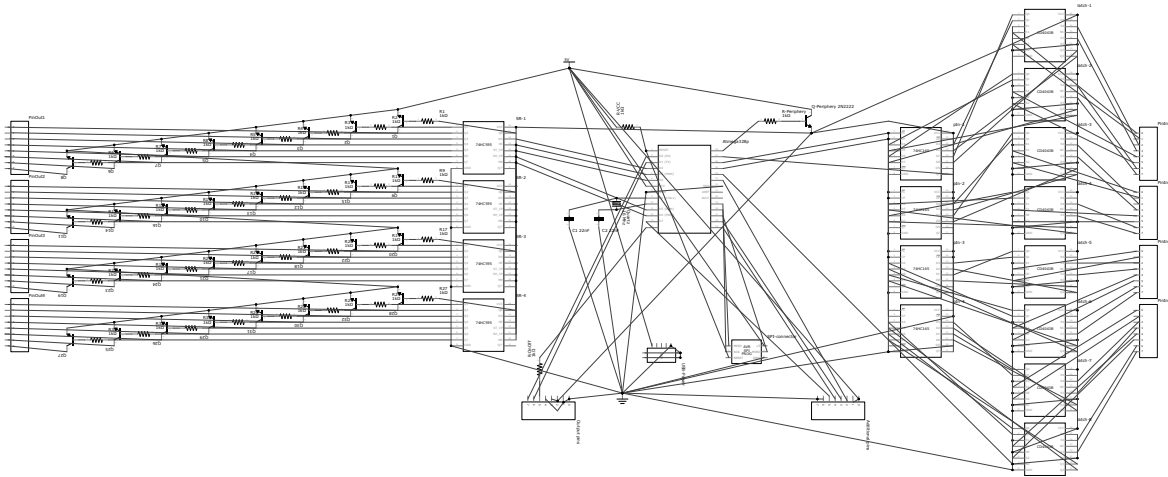


Figure 1: Schematic view of the circuit board design.

1.2 Layout - Top

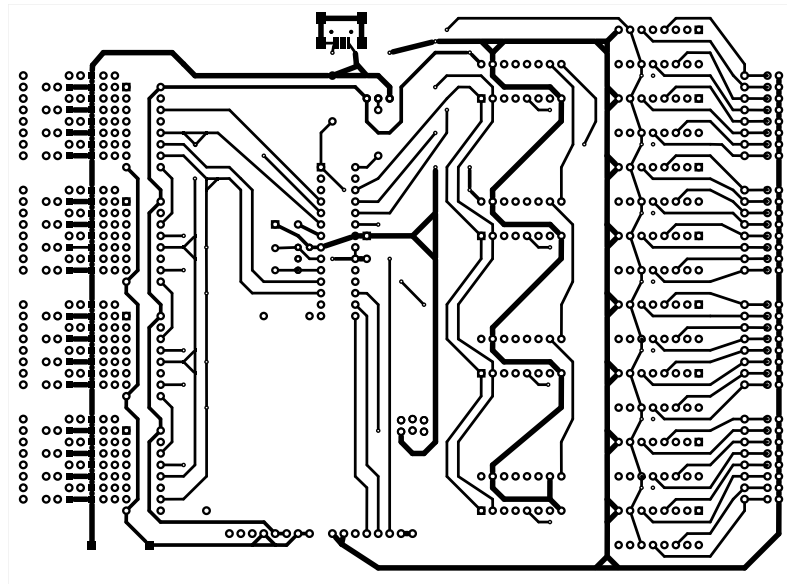


Figure 2: View of the circuit board top as seen from the top.

1.3 Layout - Bottom

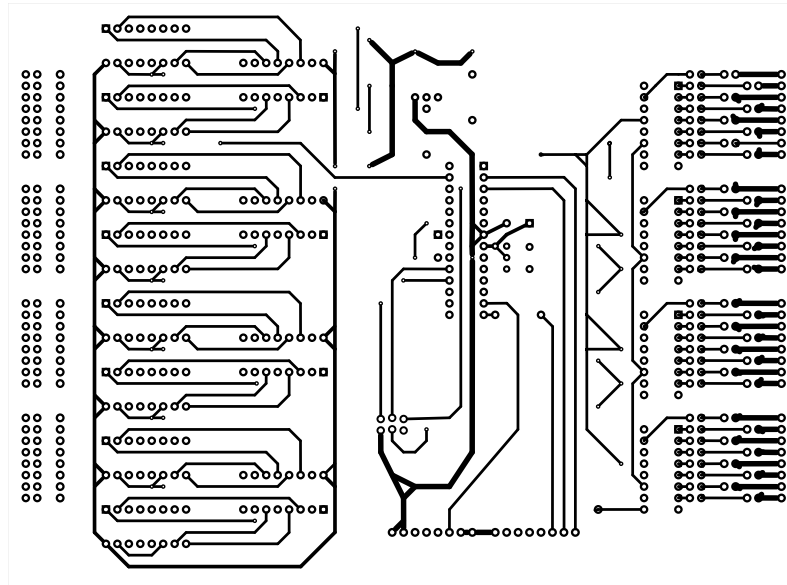


Figure 3: View of the circuit board bottom as seen from the bottom.

1.4 Layout - Hardware

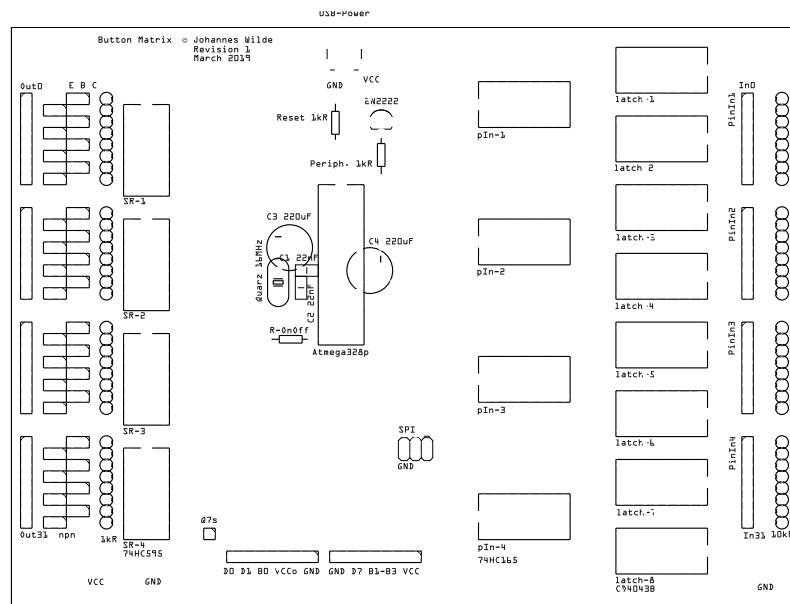


Figure 4: Positioning of the hardware as seen from the top.

1.5 Pin Connections

2 Hardware Description

2.1 ATmega328P

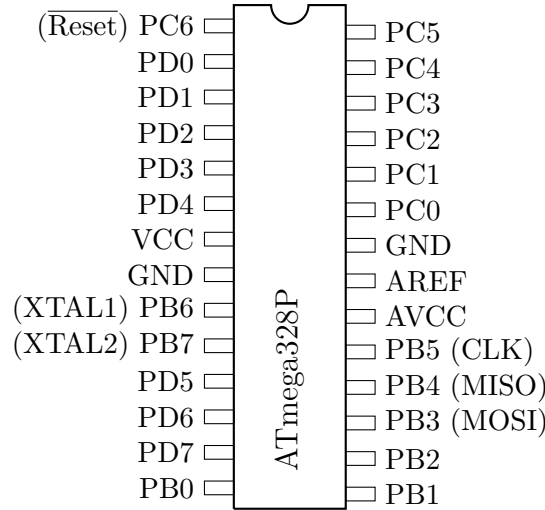


Figure 5: Schematic of the ATmega328P.

The bracketed names specify functionality assigned to the respective pins by the underlying hardware design. If one would want, these can be reconfigured for different use cases however.

2.2 74HC595 - 8-bit Shift Register

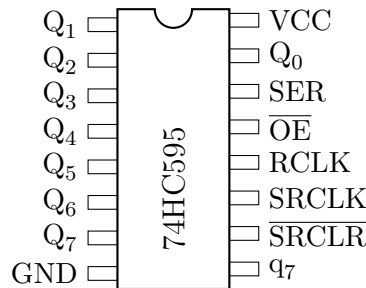


Figure 6: Schematic of the 8-bit shift register 74HC595.

Working principle:

The shiftregister has an internal $[q_i, i \in [0, 7]]$ and an external $[Q_i, i \in [0, 7]]$ 8-bit register. On a rising edge on SRCLK the values q_i are shifted to q_{i+1} [$i \in [0, 6]$, higher i first] and SER to q_0 .

On a rising edge on RCLK the values of q_i are copied to Q_i [$i \in [0, 7]$]. These will only be visible externally if \overline{OE} is LOW [otherwise the outputs will be in a high impedance state].

A LOW on \overline{SRCLR} will set q_i [$i \in [0, 7]$] LOW.

VCC and GND are required for the shift-register to work.

q_7 can be used to pass the out-shifted bits on to another shift-register, if q_7 is connected to SER of the next shift-register [which will have to be clocked accordingly].

Q_i - Output i ; GND - Ground; q_7 - Serial output; $\overline{\text{SRCLR}}$ - Clear internal shift-register on LOW; RCLK - Copy internal to external shift-register; SRCLK - shift in SER on rising edge; $\overline{\text{OE}}$ - Output Enable; SER - Serial input; VCC - Supply Voltage.

2.3 74HC165 - 8-bit parallel in, serial out register

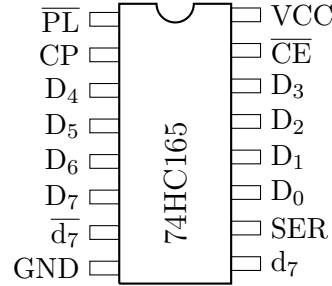


Figure 7: Schematic of the parallel in, serial out IC 74HC165.

Working principle:

The parallel load, serial out IC has an internal $[d_i, i \in [0, 7]]$ and an external $[D_i, i \in [0, 7]]$ 8-bit register.

When $\overline{\text{PL}}$ is LOW the D_i are copied to the d_i $[i \in [0, 7]]$ asynchronously, i.e. without the need for a clock.

When $\overline{\text{PL}}$ is HIGH the 74HC165 will function as a shift-register: on a positive edge on CP [if $\overline{\text{CE}}$ is low] the values d_i are shifted to d_{i+1} $[i \in [0, 6], \text{ higher } i \text{ first}]$ and SER to d_0 . Additionally the complementary signal of the new d_7 will be visible on $\overline{d_7}$.

VCC and GND are required for the 74HC165 to work.

d_7 can be used to pass the out-shifted bits on to another 74HC165, if d_7 is connected to SER of the next 74HC165 [which will need to have $\overline{\text{PL}}$ low, $\overline{\text{CE}}$ low and be clocked accordingly].

$\overline{\text{PL}}$ - Not Parallel Load; CP - Clock; D_i - Parallel Data In; d_7 - Serial Output; $\overline{d_7}$ - Not Serial Output; GND - Ground; SER - Serial In; $\overline{\text{CE}}$ - Not Clock Enable; VCC - Supply Voltage.

2.4 CD4043B - CMOS Quad 3-State R/S-Latches [NOR]

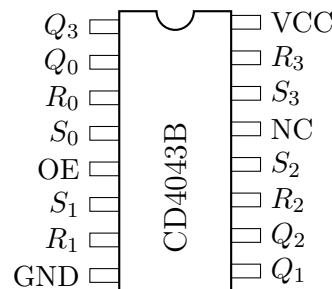


Figure 8: Schematic of the quad 3-state R/S Latch CD4043B.

Working principle:

OE controls whether the outputs Q_i $[i \in [0, 3]]$ are connected [OE is HIGH] or in a high-

Table 1: Functional states of the quad 3-state R/S Latch CD4043B [$i \in [0, 3]$].

S_i	R_i	OE	Q_i
X	X	LOW	high impedance
LOW	LOW	HIGH	unchanged
HIGH	LOW	HIGH	HIGH
LOW	HIGH	HIGH	LOW
HIGH	HIGH	HIGH	HIGH

impedance state [like an open circuit; OE is LOW].

Each separate output Q_i [$i \in [0, 3]$] can be set to HIGH [S_i HIGH] or reset to LOW [R HIGH] separately and asynchronously.

In case S_i and R_i are HIGH simultaneously, the respective output Q_i will read HIGH.

NC is not connected internally.

Q_i - Output i ; R_i - Reset i ; S_i - Set i ; OE - Output Enable; GND - Ground; NC - Not Connected; VCC - Supply Voltage.

2.5 LED

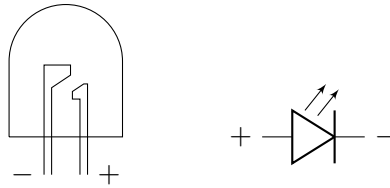


Figure 9: Schematic view of a LED and its representation in circuit diagrams.

2.6 npn-Transistor

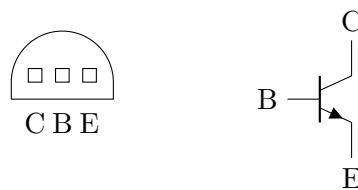


Figure 10: Schematic top view onto a npn-transistor and its representation in circuit diagrams.

2.7 Periphery

3 SPI-Programming

The ATmega328P can be programmed using the serial peripheral interface [SPI].

I used an Arduino Uno programmed with the “Arduino ISP” sketch [under Examples in the Arduino IDE] to program my separate ATmega328P.

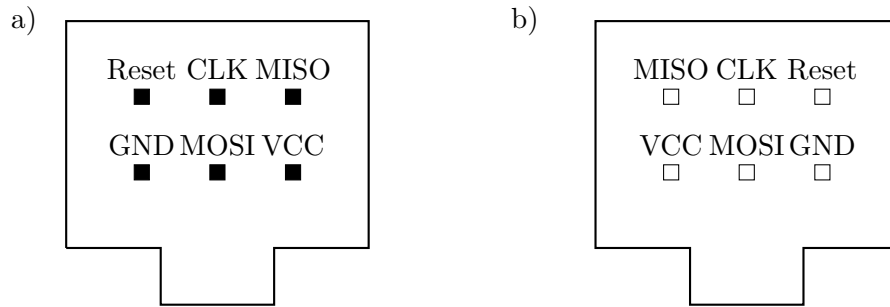


Figure 11: SPI connectors - a) connector, b) cable.

In order for that to work one will have to keep the Reset pin of the programming Arduino Uno HIGH with a capacitor and connect the Reset pin of the device to be programmed with the pin as specified in the “Arduino ISP” sketch via `#define RESET 10` [this being pin D10 per default].

Other than that each pin of the programmer will have to be connected with the pin of the same name of the programme [i.e. e.g. MOSI-programmer → MOSI-programmee].