Zu Beginn wird das Schieberegister zurückgesetzt. Danach werden die einzelnen Bits der Zahl 21 mit jedem Takt an das Schieberegister übertragen. Dabei ist darauf zu achten, dass die Zeit zwischen den Signalwechsel lang genug ist.

```
/* ET_48_74HC164_Test
 *
 * Pinbelegung
 *
 * Funktion | 74164  | Arduino | MC
 * Neg CLR  | 9      | A1      | PC0
 * Data     | 1, 2   | A2      | PC1
 * CLK      | 8      | A3      | PC2
 */

#define PIN_NEG_CLR 0
#define PIN_DATA 1
#define PIN_CLK 2
#define THIRD_CLK_MS 10

// 21 entspricht 0100 1000
const byte data = 0x48;

void transmit_data() {
  PORTC &= ~(1 << PIN_NEG_CLR);
  delay(THIRD_CLK_MS);

  PORTC |= 1 << PIN_NEG_CLR;
  delay(THIRD_CLK_MS);

  for (uint8_t bit_index = 0; bit_index < 8; bit_index++) {
    // Serial data bit
    const byte data_bit = (data >> bit_index) & 0x01;

    // Clear PIN_DATA then set the data_bit
    PORTC = (PORTC & ~(1 << PIN_DATA)) | data_bit << PIN_DATA;
    delay(THIRD_CLK_MS);

    // Clock high
    PORTC |= 1 << PIN_CLK;
    Serial.println("Clock high");
    delay(THIRD_CLK_MS);

    // Clock low
    PORTC &= ~(1 << PIN_CLK);
    Serial.println("Clock low");
    delay(THIRD_CLK_MS);
  }

  PORTB |= 1 << PB5;
}

void setup() {
  Serial.begin(9600);

  // Use onboard LED
  DDRB = 1 << PB5;
```

```
53
54    DDRC = 0x07;
55    PORTC = 1 << PIN_NEG_CLR;
56
57    transmit_data();
58  }
59
60  void loop() {
61  }
```

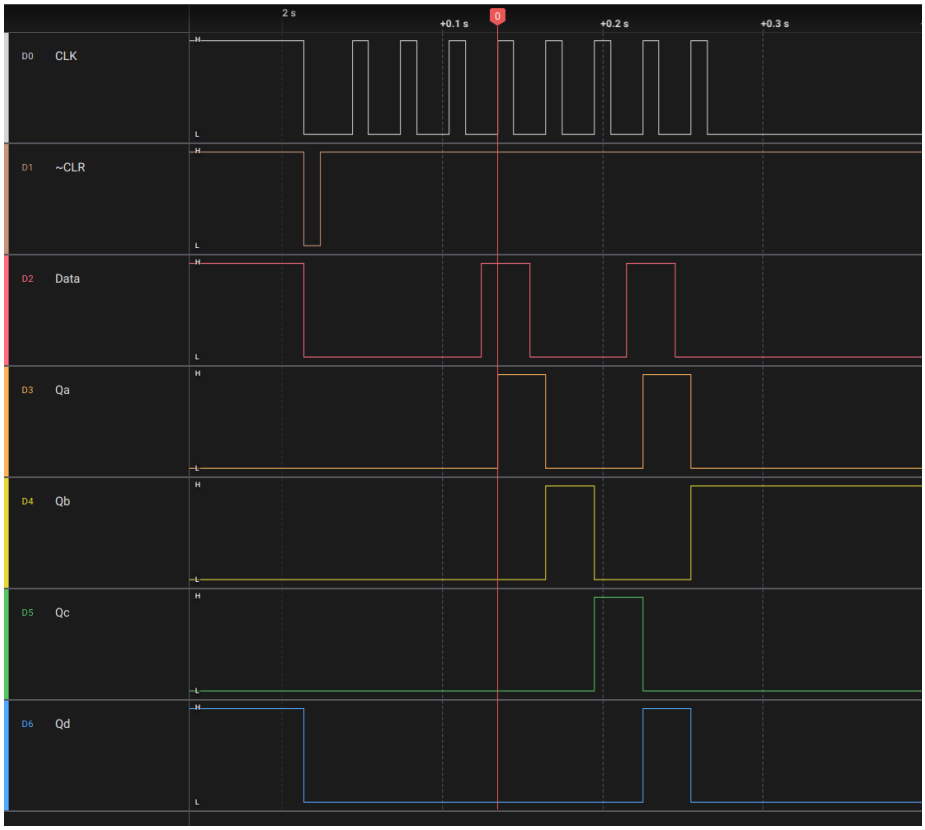

Abbildung 1: Messung von Qa bis Qd

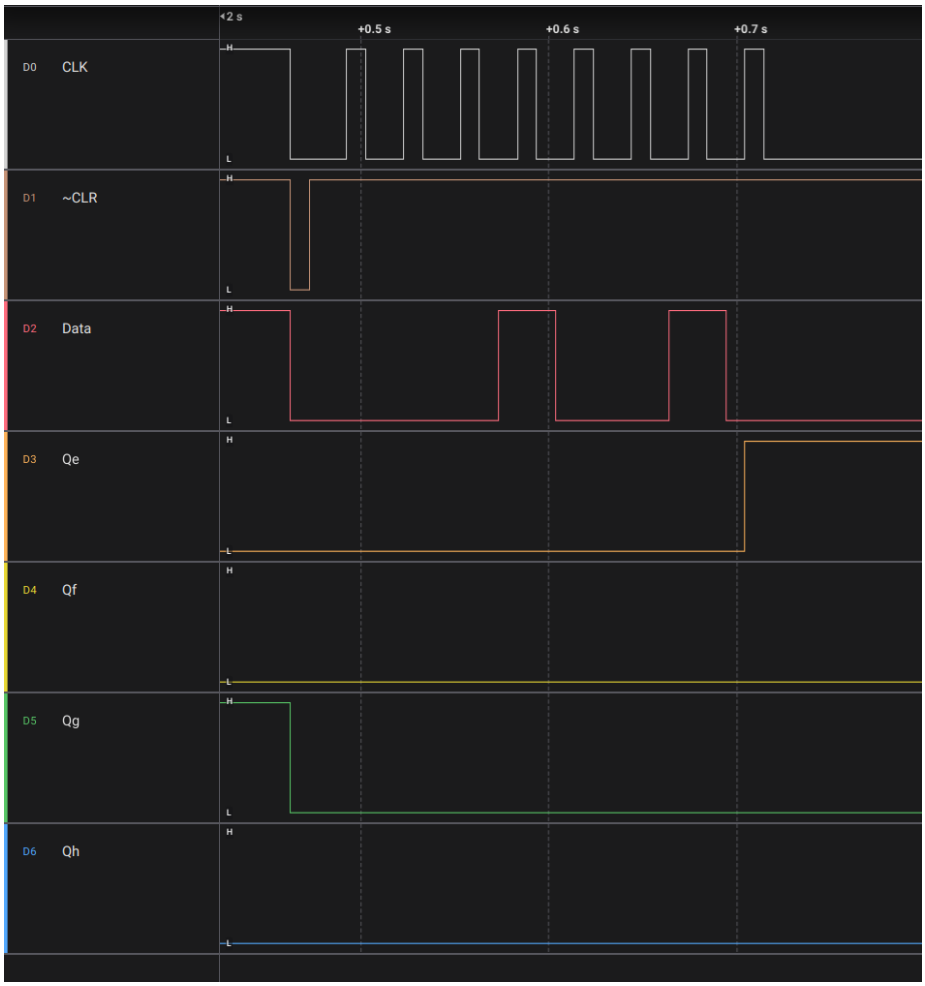

Abbildung 2: Messung von Qe bis Qh