

# Protokoll des I<sup>2</sup>C Test auf dem ml605

Nach dem Start des Testprogramms meldet sich dieses auf der Konsole wie folgt:

**Test des IIC Master und Slave vom SpartanMC auf ml605**

**Bei Eingabe von ESC wechsel von Test1 und Test2**

**Im Test 1 wechsel der Bitkombination bei jeder anderen Eingabe**

**Einstellung der Datenrate**

**1 - 100,000kHz**

**2 - 400,000kHz**

**3 - 1015,385kHz**

**4 - 2200,000kHz**

**5 - 2640,000kHz**

**6 - 3300,000kHz ? : 6**

**Send 2, 3, 5, 7 oder 8 Byte ? : 2**

**Test 1**

**write 2 Byte to device and read 3 Byte from device successful 0**

**write buf: D7 D8 D9 DA**

**read buf: 55 D7 D8 00**

**slave buf: D7 D8 DD DE**

**RxInt buf: 04104 04105 04106 00000 00000 00000 00000 00000 00000 00000**

**TxInt buf: 04500 04501 04502 04503 00000 00000 00000 00000 00000 00000**

**Test 2**

**write 2 Byte to device and read 3 Byte from device successful 0**

**write buf: 96 97 98 99**

**read buf: 55 96 97 00**

**slave buf: 96 97 9C 9D**

**RxInt buf: 04104 04105 04106 00000 00000 00000 00000 00000 00000 00000**

**TxInt buf: 04500 04501 04502 04503 00000 00000 00000 00000 00000 00000**

**Fehler 0 Fehler 0 Es sind keine Fehler aufgetreten!**

Mit diesen Einstellungen wurden dann die folgenden Bilder mit dem Oszilloskop beim senden an den Slave mit der Adresse 0x12 mit dem Byte 0xd7 und 0xd8 aufgenommen. In obigen Protokoll stehen im „write buf“ als erstes die beiden Bytes, die an den Slave gesendet wurden. Im „slave buf“ müssen an den ersten beiden Positionen die gleichen Byte stehen. Im „read buf“ steht als erstes immer das Byte mit dem das Slave Tx Register beim Start des Programm Zyklus initialisiert war gefolgt von den beiden gesendeten Bytes. In den Zeilen „RxInt buf“ und „TxInt buf“ wird bei jedem Interrupt des Masters der Inhalt des Statusregisters in den Bits 15 bis 8 und die Nummer des Interrupts in den Bits 7 bis 0 angezeigt. Man sieht, das es 4 Tx Interrupts (Senden Adr. 0x12, 0xd7, 0xd8 und erneut die Adr. 0x12) und 3 Rx Interrupts gibt (Empfang der Byte 0x55, 0xd7 und 0xd8).

Das folgende 1. Bild zeigt den vollständigen Datentransfer mit dem Senden von 2 Bytes und dem Empfangen von 3 Bytes. In den folgenden Bildern wird dann das Signalspiel für jedes einzelne Byte dieser Nachricht im Zoom dargestellt.

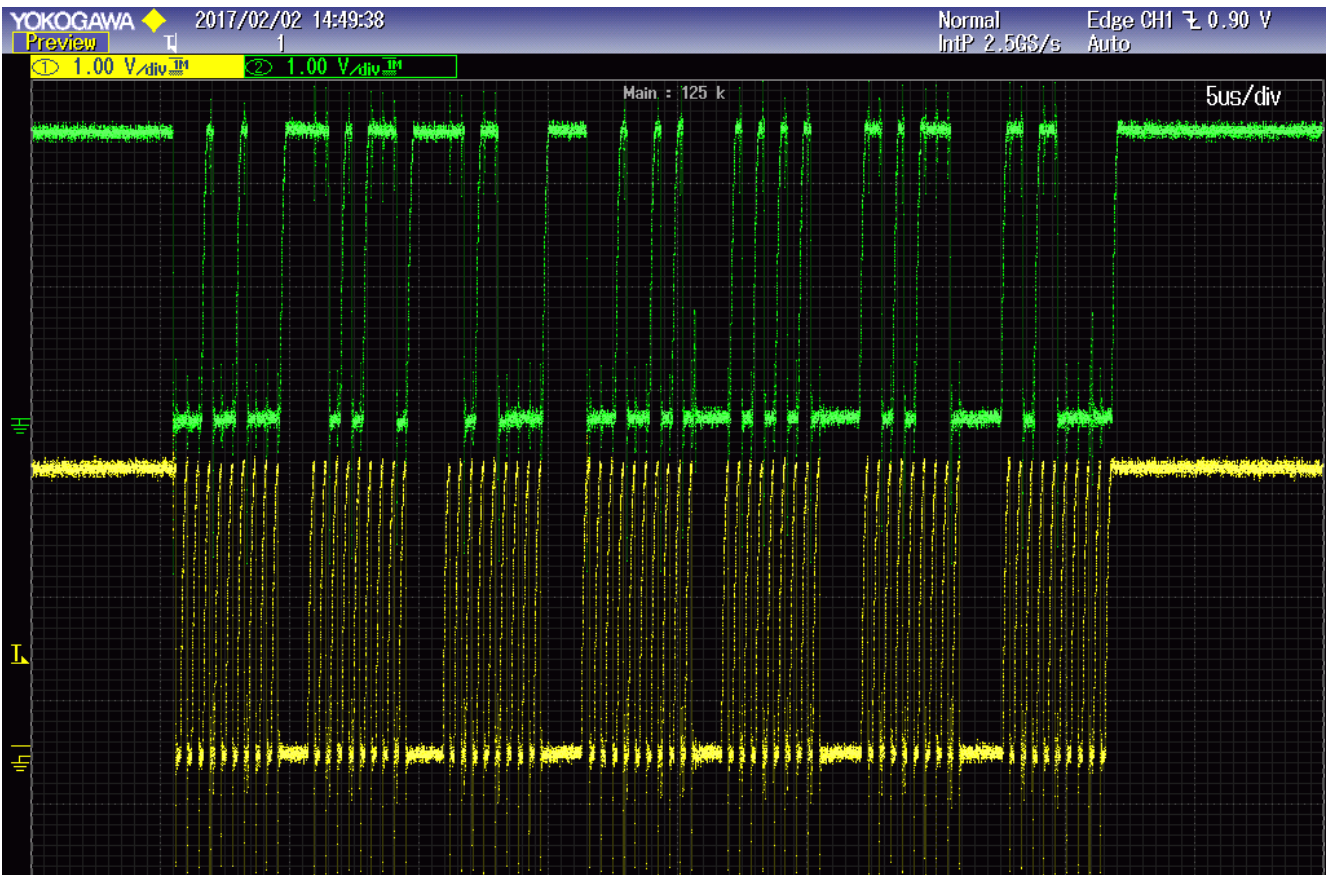


Bild 1: Senden von 0xd7 und 0xd8 an den Slave 0x12 und empfangen der Byte 0x55, 0xd7 und 0xd8 vom Slave 0x12.



Bild 2: Senden der Adresse 0x12 mit dem WR-Flag und ACK. Die Übertragung beginnt mit einem Start.

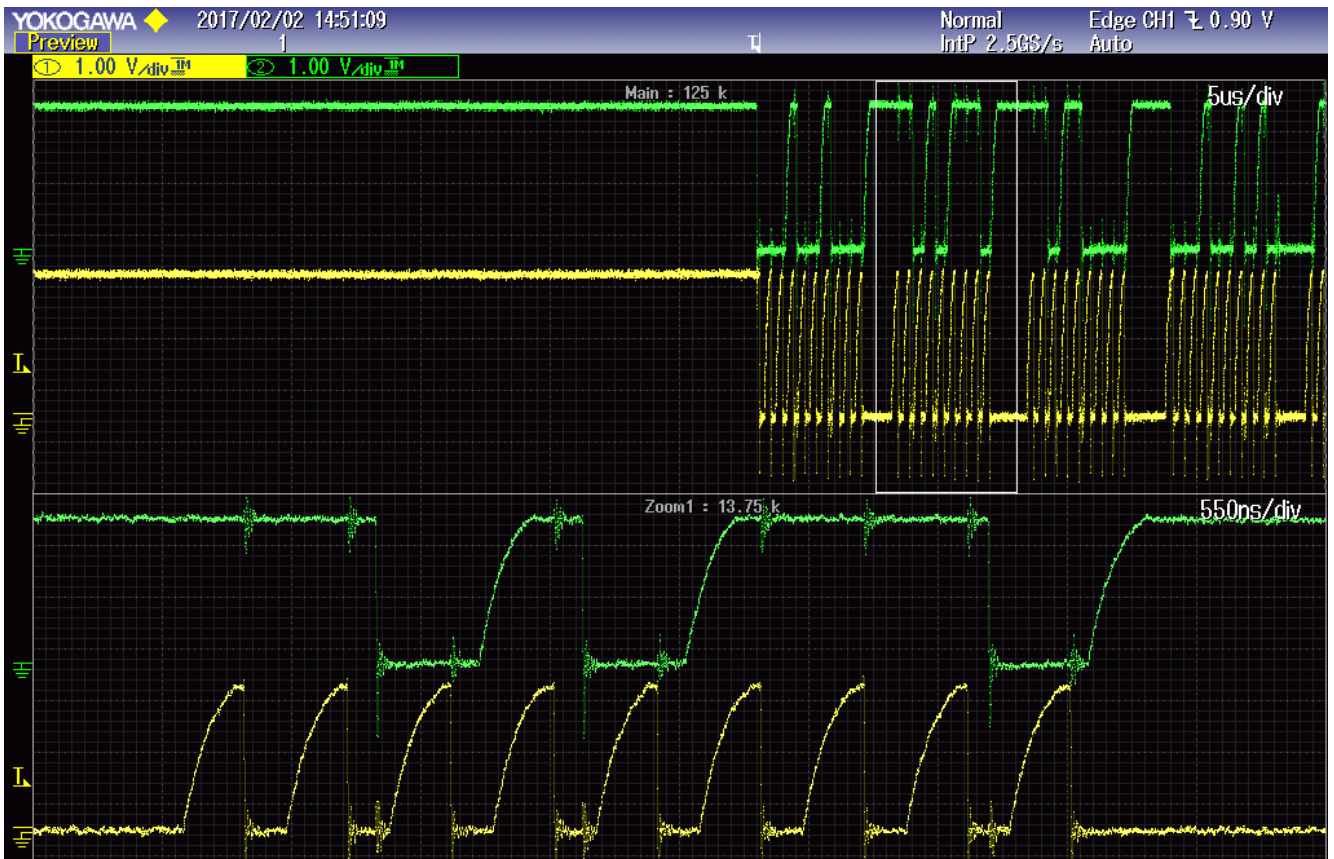


Bild 3: Senden des Datenbyte 0xd7 und ACK als letzte Bit.

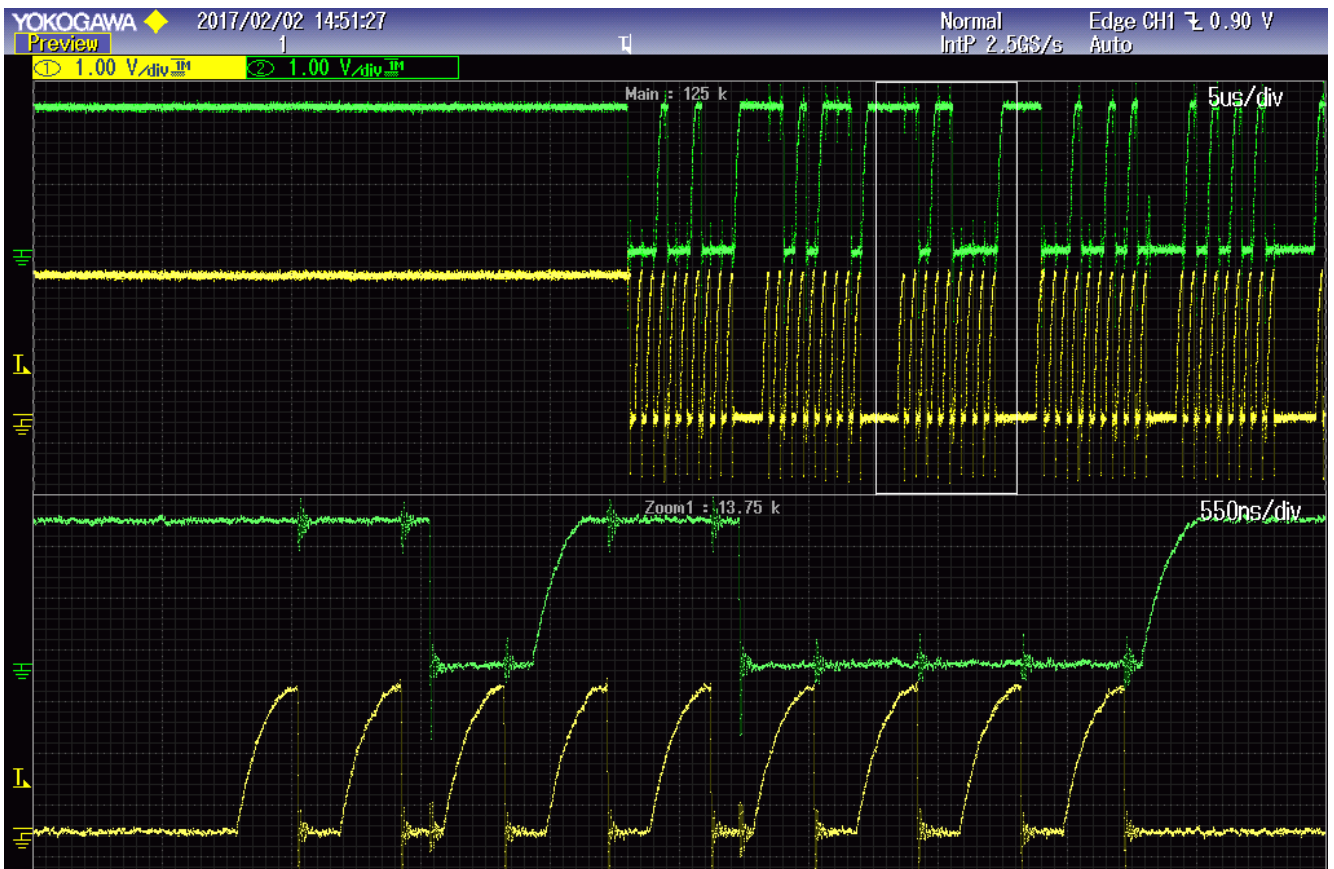


Bild 4: Senden des Datenbyte 0xd8 und ACK als letztes Bit gefolgt von einem erneutem Start für den Empfang der 3 Bytes.

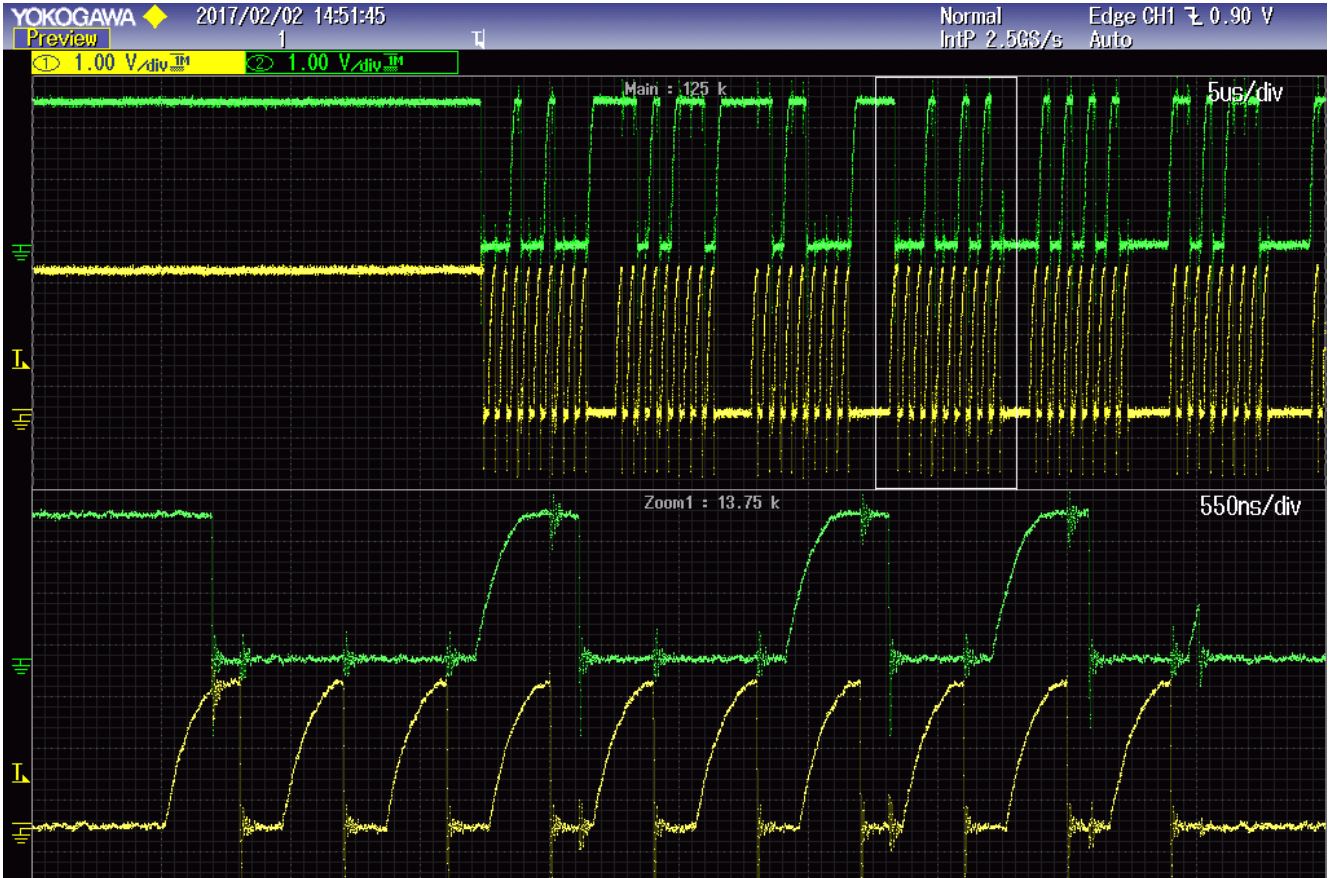


Bild 5: Start und erneutes Senden der Adresse 0x12 aber dieses mal mit dem Read-Flag und ACK.

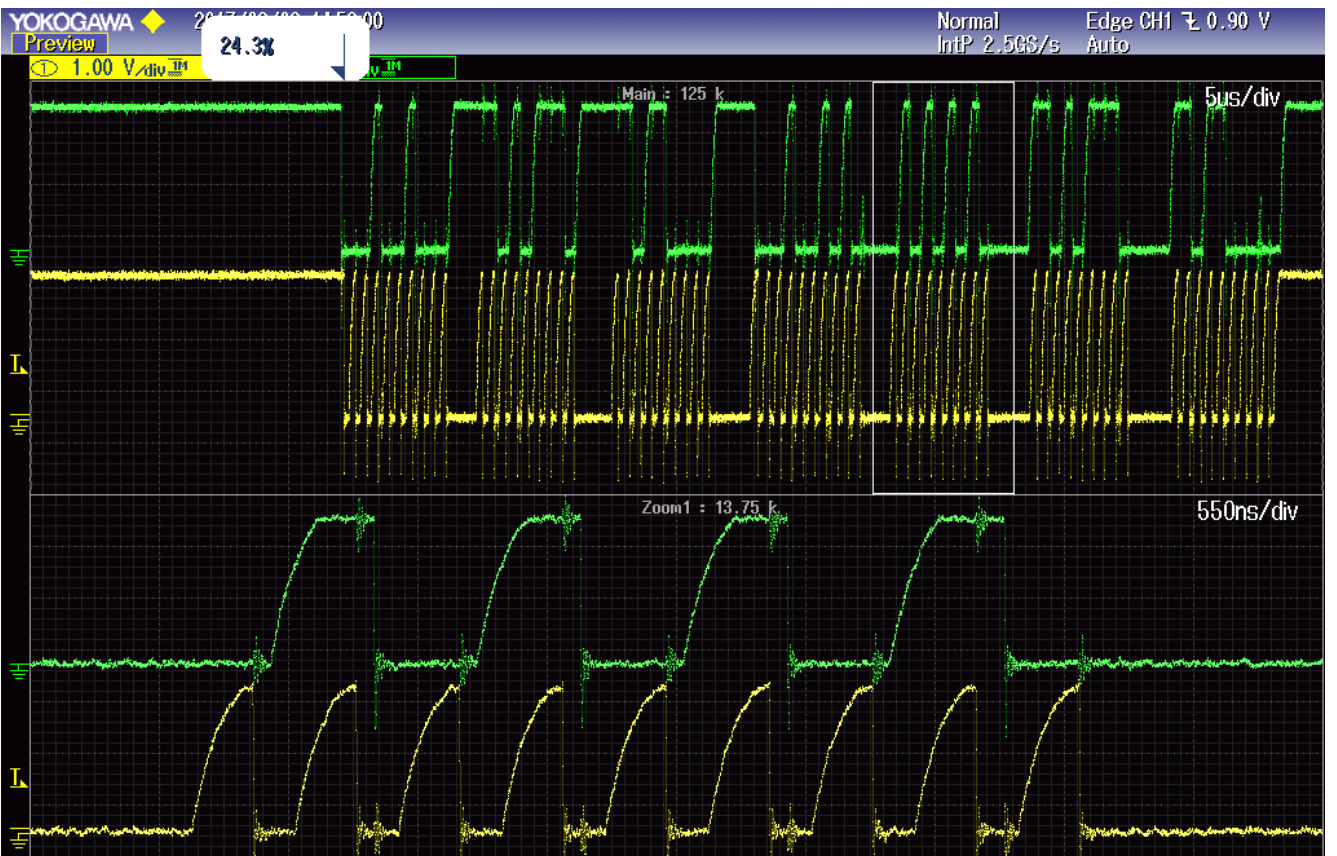


Bild 6: Empfangen des 1. Bytes 0x55 und ACK.

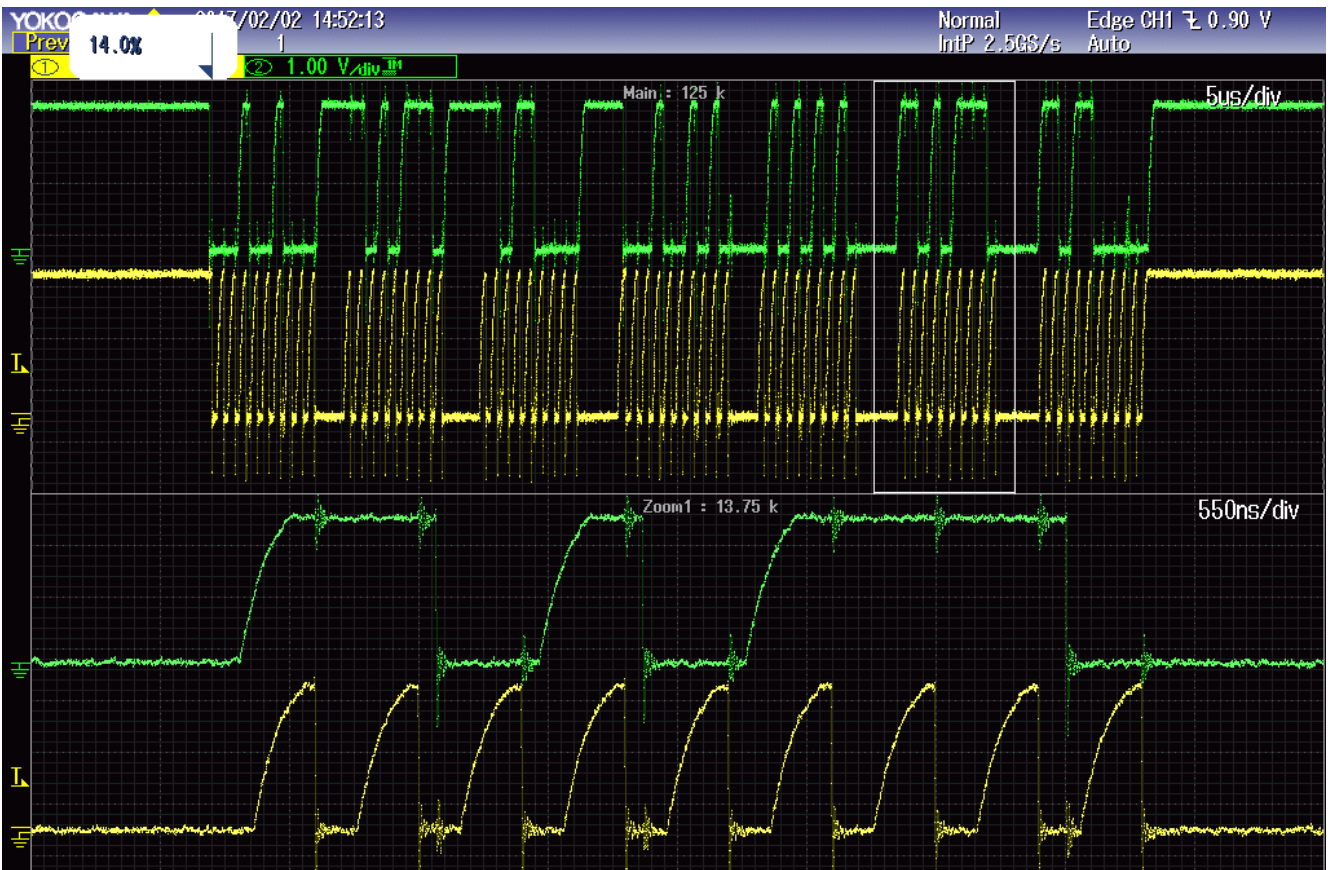


Bild 7: Empfangen des 2. Datenbytes 0xd7 und Senden von ACK.

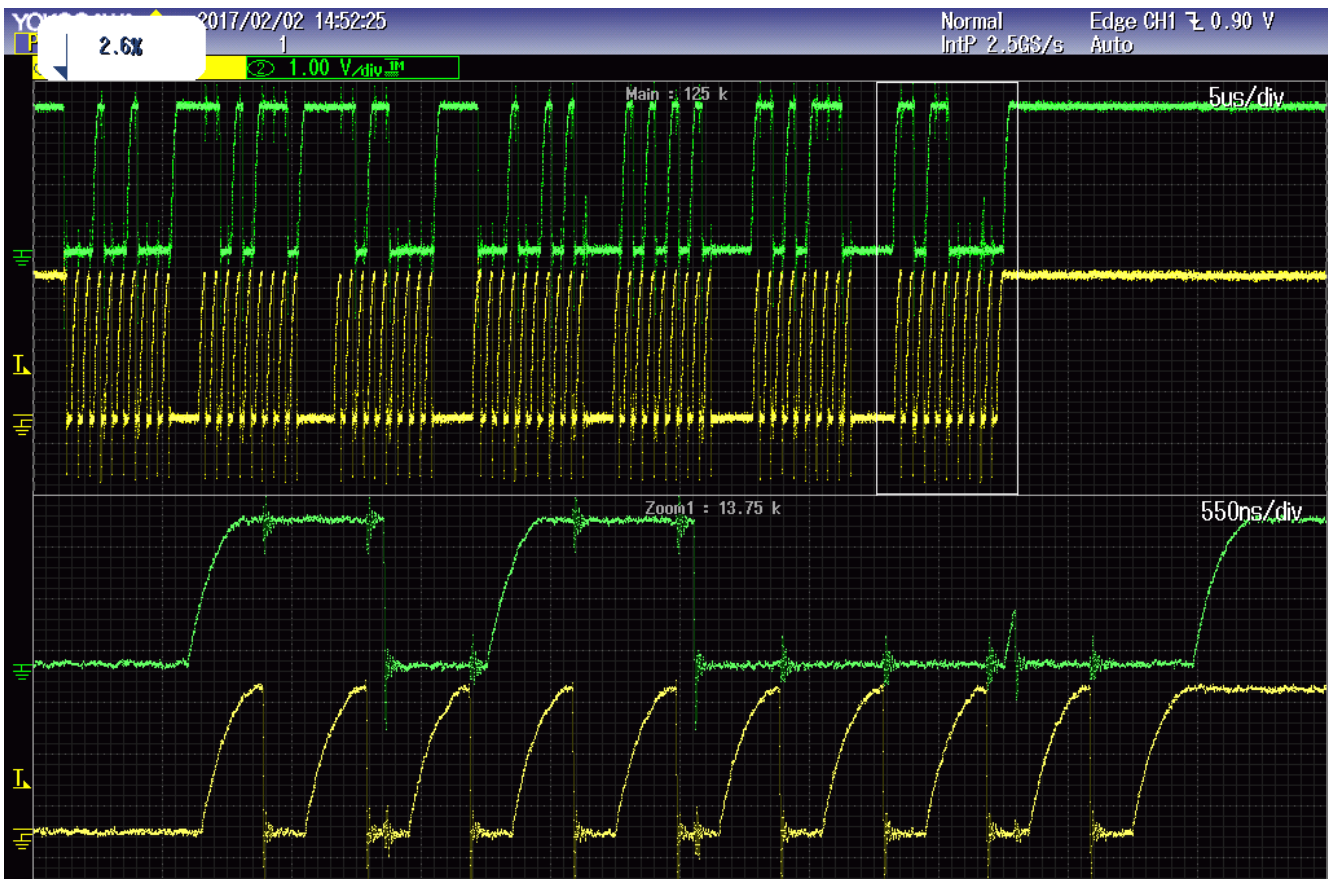


Bild 8: Empfangen des letzten Bytes 0xd8 und Senden von ACK gefolgt vom Stopp Befehl.

