TPS65381 AMUX (Analog Multiplex) parameters out from DIAG_OUT pin at VBATP=12V and 14V

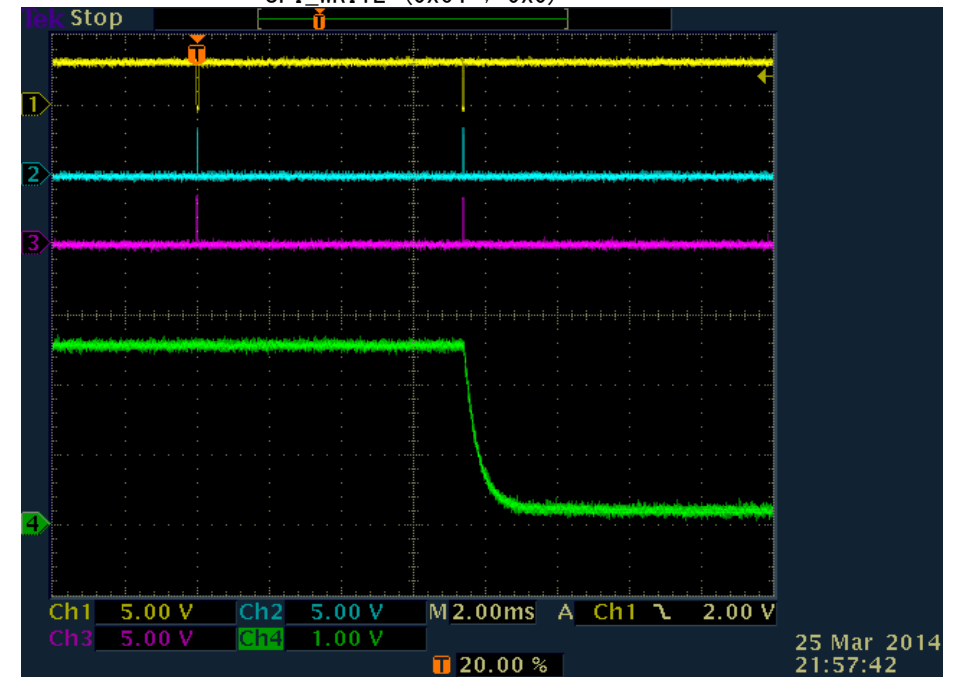|	Voltage Rail / Signal Name	Description	Divide Ratio	Voltage range / Accuracy	Measured Voltage @VBATP=12V	Measured Voltage @VBATP=14V
A.1	VDD5	Linear VDD5 Regulator output	2±1.5%	2.5V±2%	2.511	2.512
A.2	VDD6	Switching-Mode Pre-Regulator	3±2.2%	2V±5%	2.012	2.012
A.3	VCP	External Charge Pump	13.5±2%	0.6V … 4V	1.564	1.845
A.4	VSOUT1	Sensor Supply Output Voltage	4±0.5%	0.825 … 2.375 ±2%	1.245	1.245
A.5	VBAT_SAFING	Safing Battery Supply	10±2%	0.4V … 4V	1.206	1.409
A.6	VBATP	Battery Supply	10±2%	0.4V … 4V	1.206	1.409
A.7	MAIN_BG	Regulators Main Bandgap Reference	1	2.5V±2%	2.498	2.496
A.8	VMON_BG	Voltage Monitor Bandgap	1	2.5V±2%	2.494	2.495

AMUX -> DIAG_OUT waveform without series resistor (BENCH + TIGER board + GUI software)
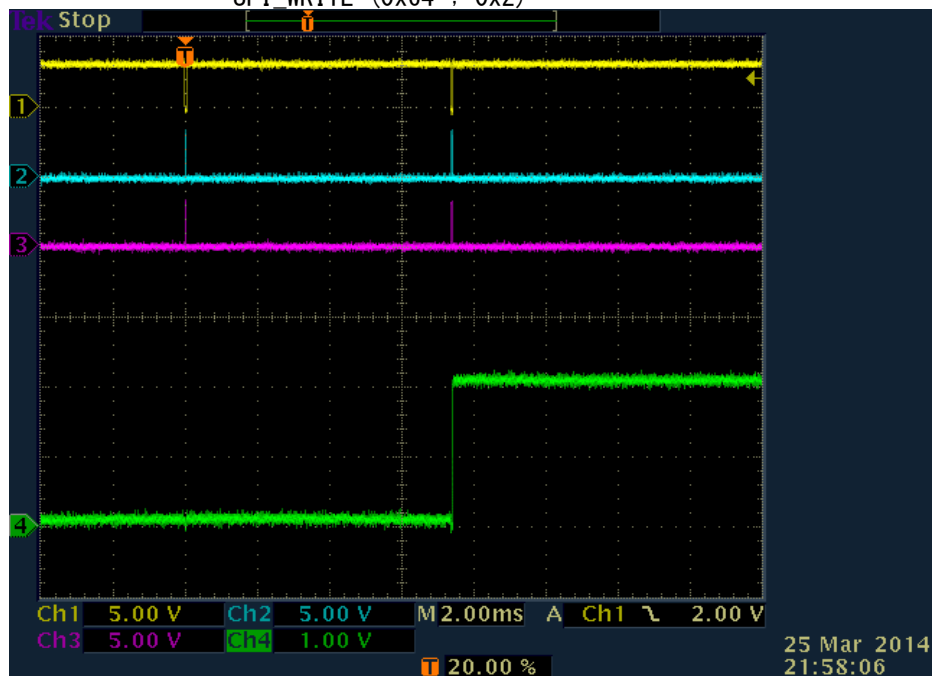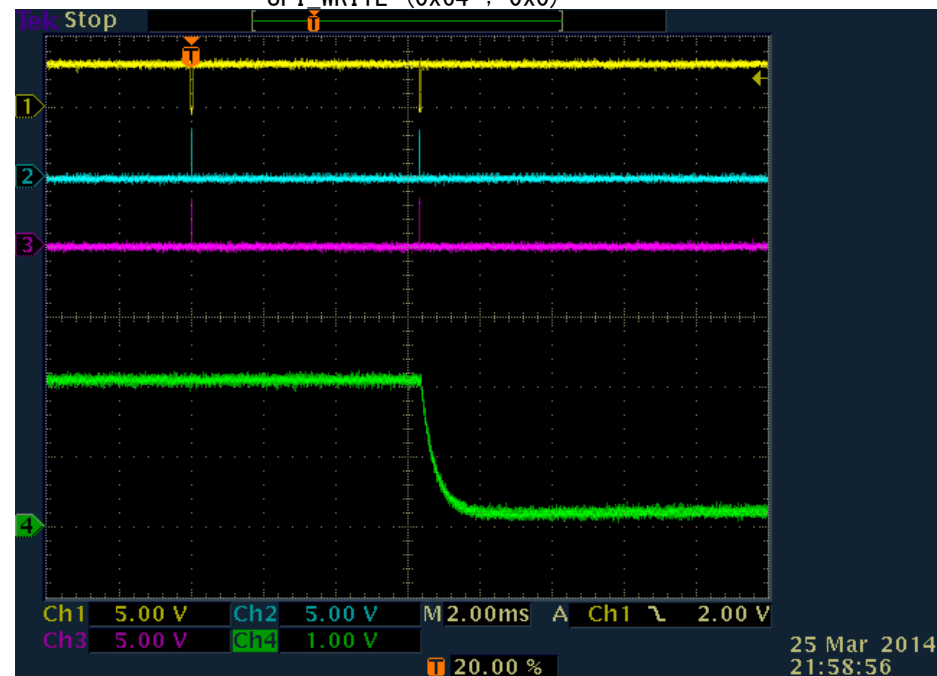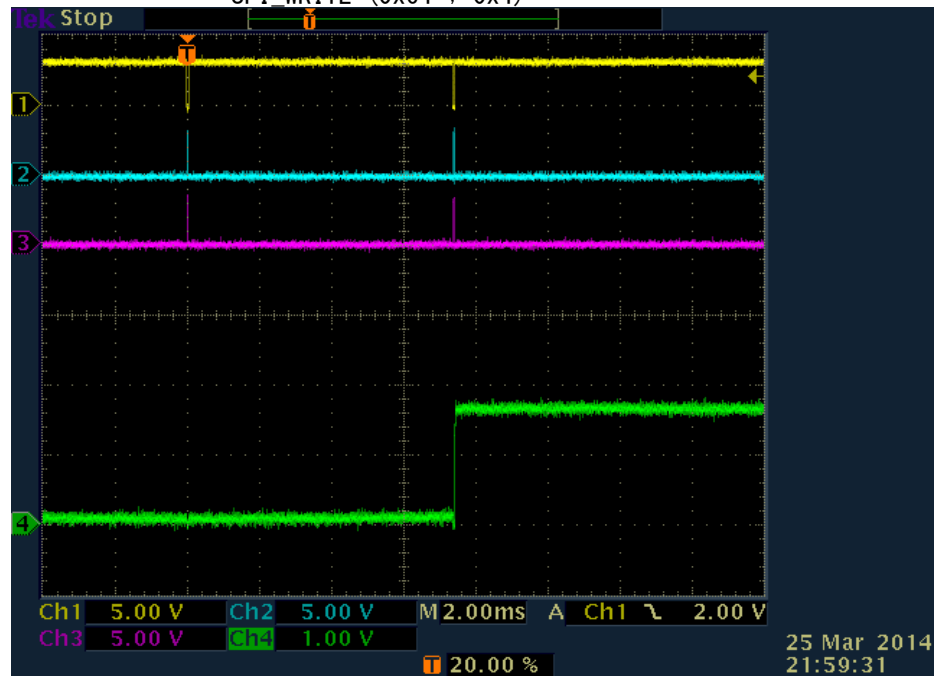
A.1 VDD5

SPI_WRITE (0x66 ; 0xa2)
SPI_WRITE (0x64 ; 0x1)

SPI_WRITE (0x66 ; 0xa2)
SPI_WRITE (0x64 ; 0x0)

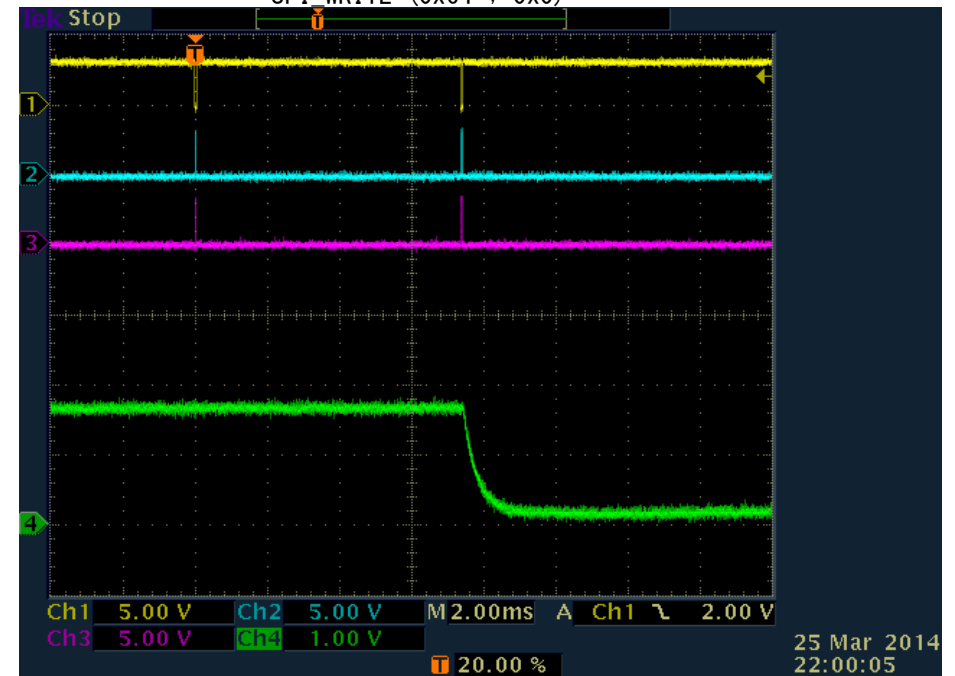AMUX -> DIAG_OUT waveform without series resistor (BENCH + TIGER board + GUI software)
A.2 VDD6   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
                SPI_WRITE (0x64 ; 0x2)   SPI_WRITE (0x64 ; 0x0)

AMUX -> DIAG_OUT waveform without series resistor (BENCH + TIGER board + GUI software)
A.3 VCP   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
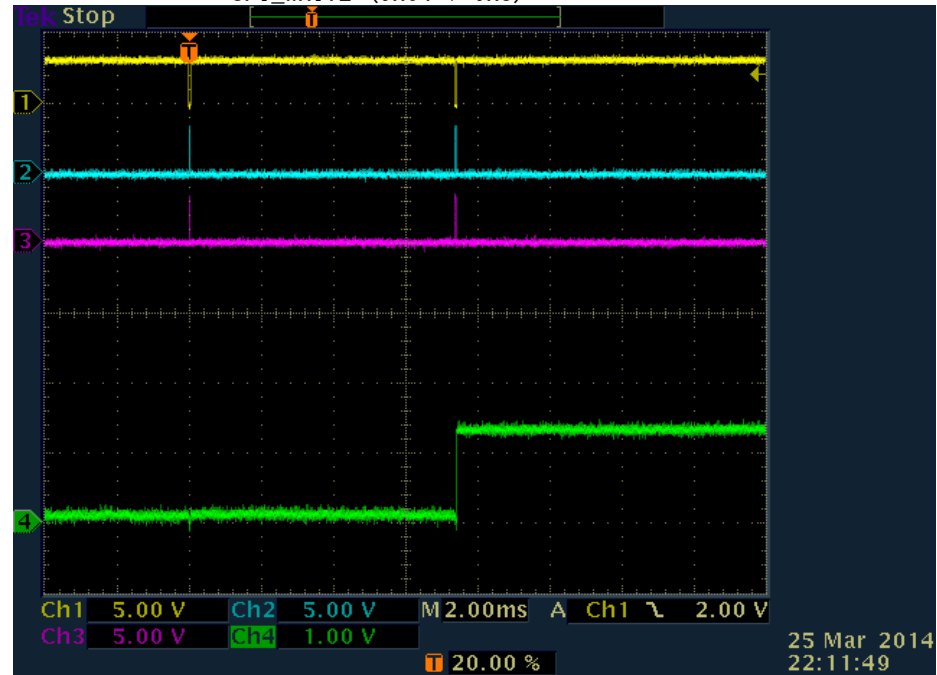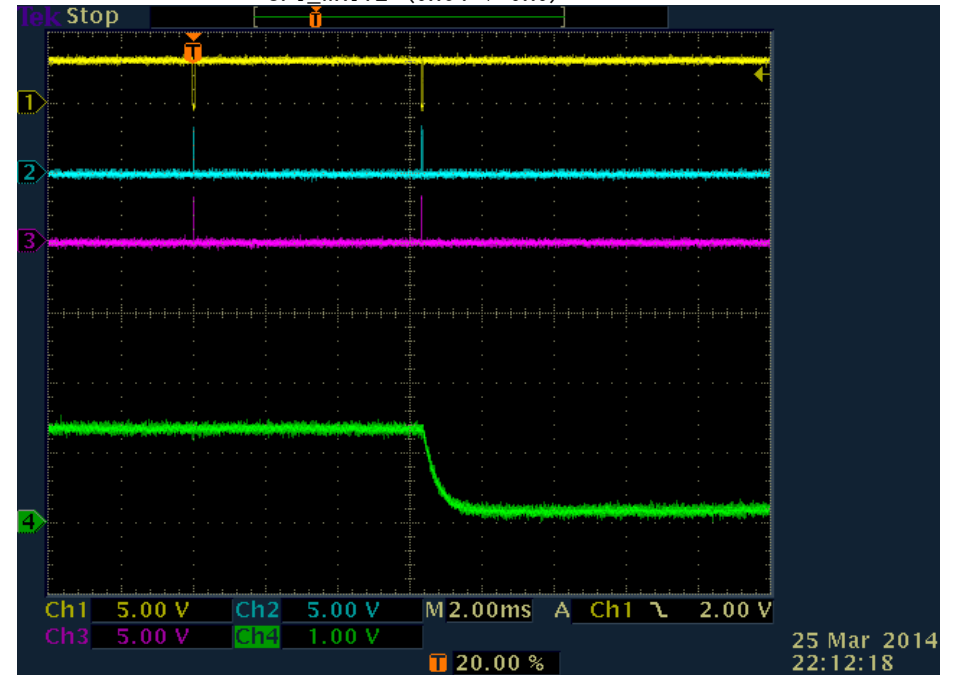   SPI_WRITE (0x64 ; 0x4)   SPI_WRITE (0x64 ; 0x0)

AMUX -> DIAG_OUT waveform without series resistor (BENCH + TIGER board + GUI software)
A.4 VSOUT1   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
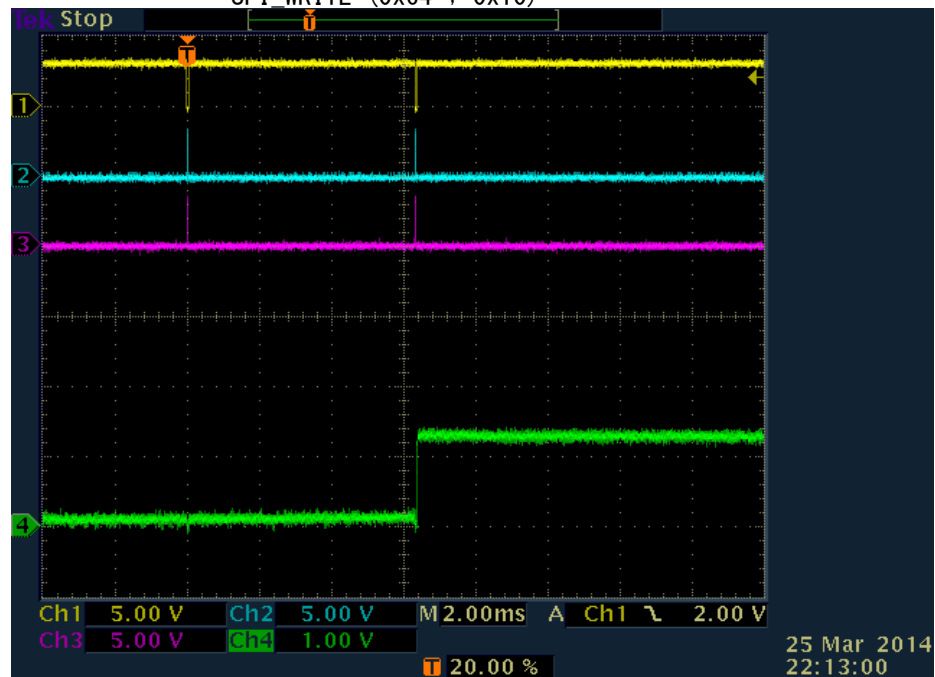   SPI_WRITE (0x64 ; 0x8)   SPI_WRITE (0x64 ; 0x0)

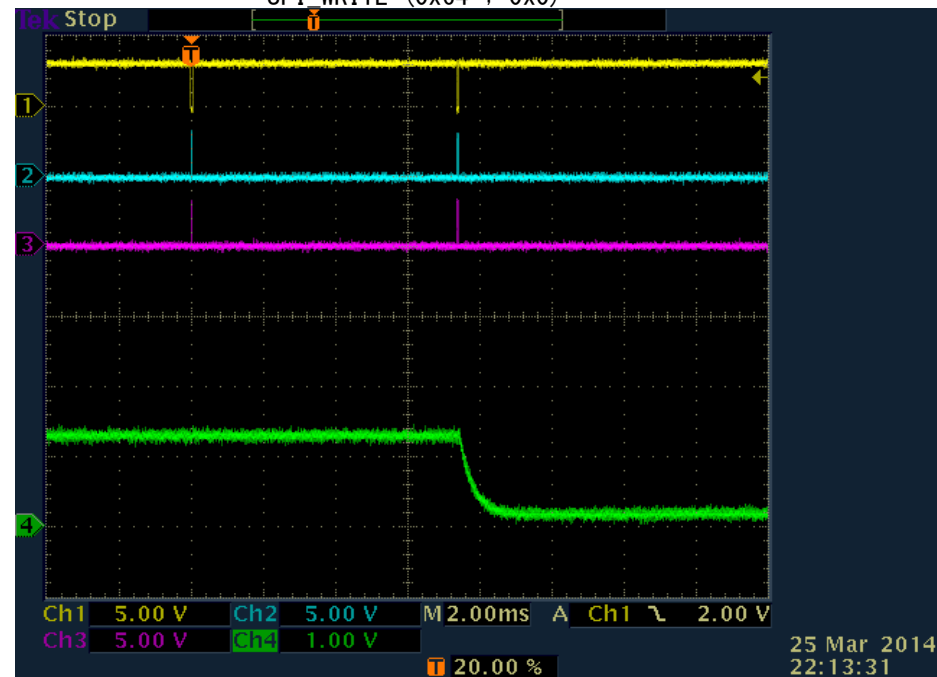AMUX -> DIAG_OUT waveform without series resistor (BENCH + TIGER board + GUI software)
A.5 VBAT_SAFING SPI_WRITE (0x66 ; 0xa2)
SPI_WRITE (0x64 ; 0x10)

SPI_WRITE (0x66 ; 0xa2)
SPI_WRITE (0x64 ; 0x0)

AMUX -> DIAG_OUT waveform without series resistor (BENCH + TIGER board + GUI software)
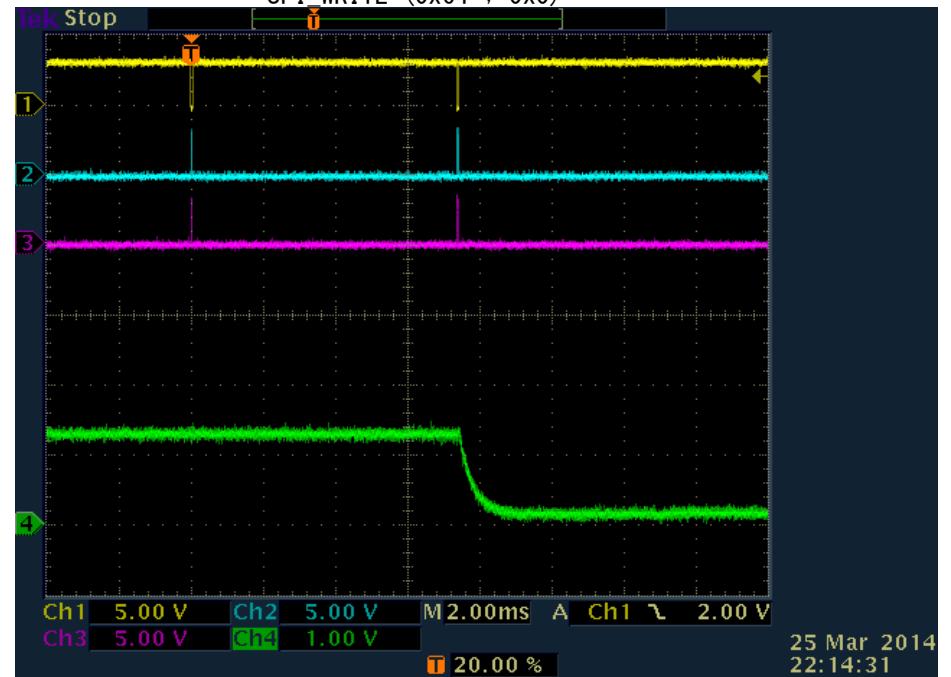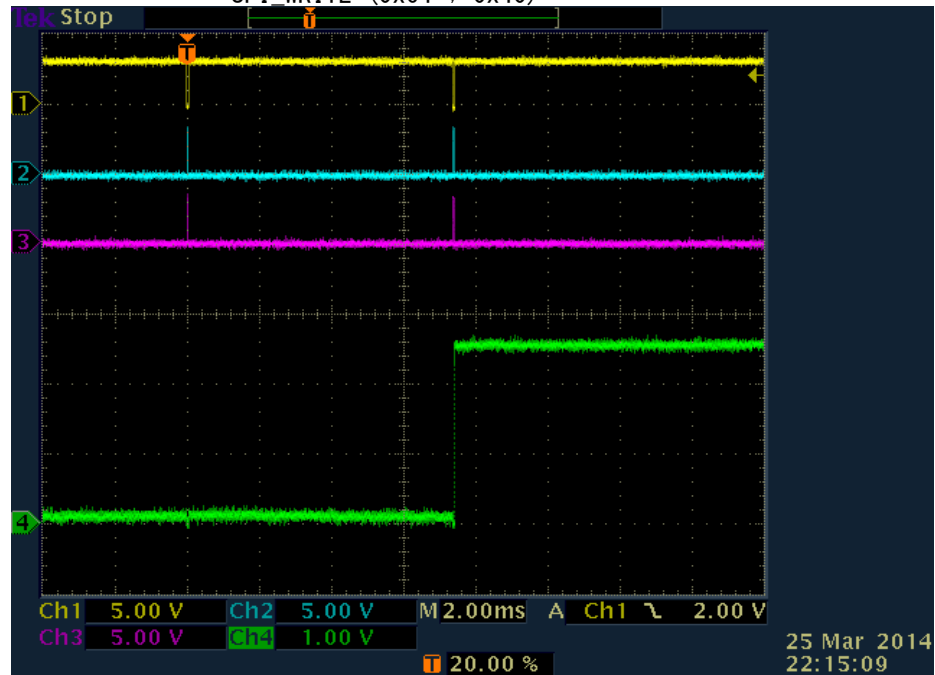A.6 VBATP   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
               SPI_WRITE (0x64 ; 0x20)   SPI_WRITE (0x64 ; 0x0)
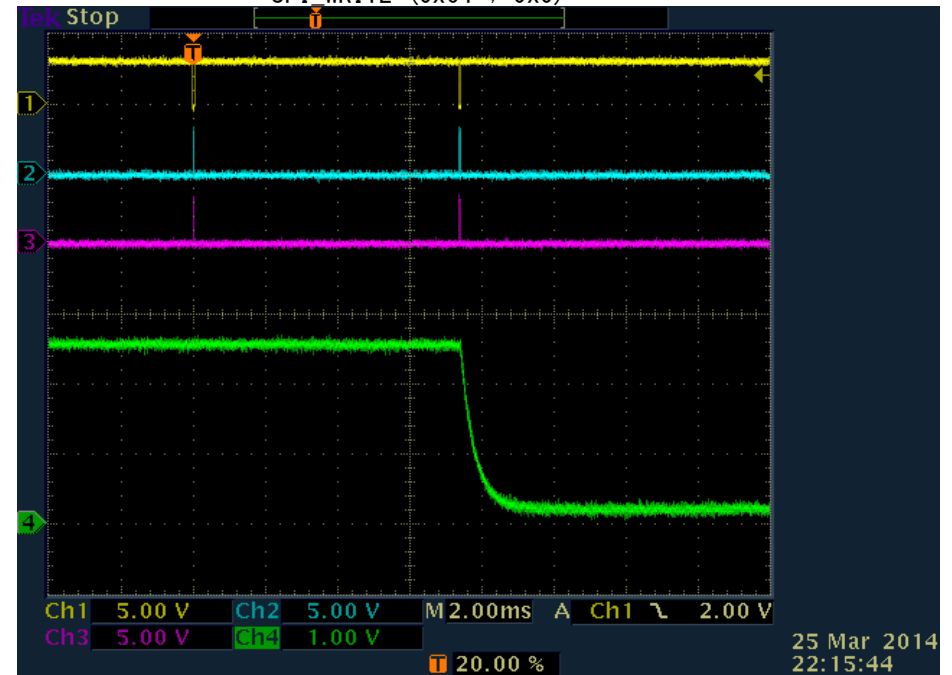
AMUX -> DIAG_OUT waveform without series resistor (BENCH + TIGER board + GUI software)
A.7 MAIN_BG   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
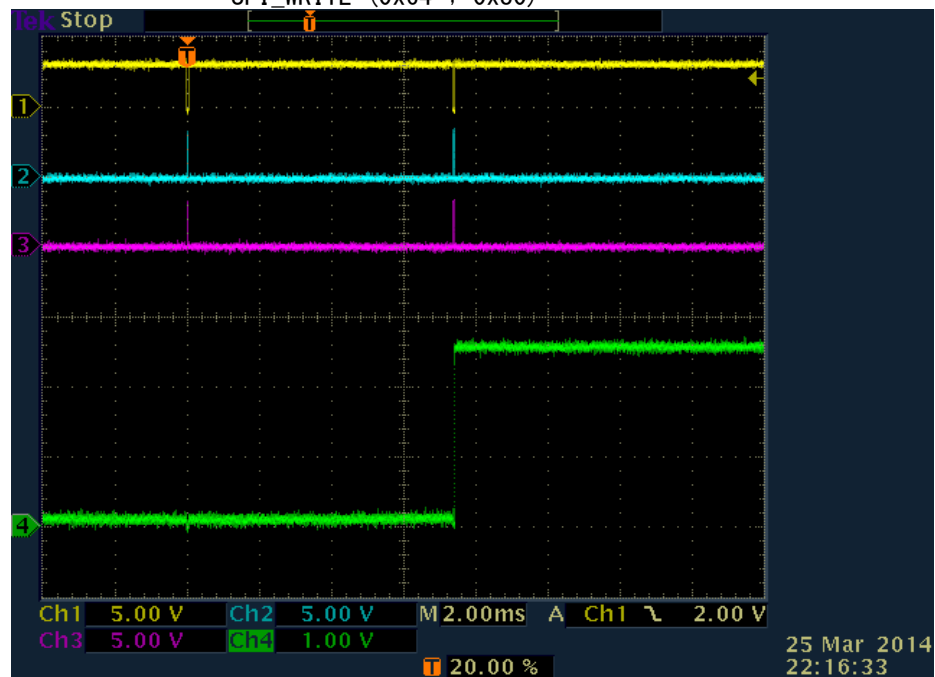   SPI_WRITE (0x64 ; 0x40)   SPI_WRITE (0x64 ; 0x0)

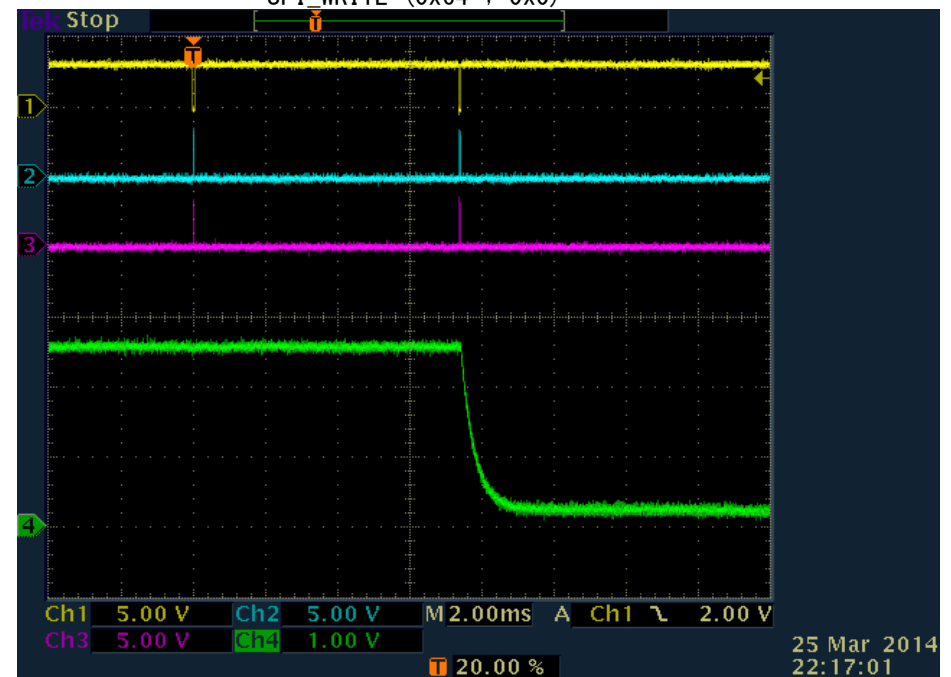AMUX -> DIAG_OUT waveform without series resistor (BENCH + TIGER board + GUI software)
A.8 VMON_BG   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
                 SPI_WRITE (0x64 ; 0x80)   SPI_WRITE (0x64 ; 0x0)
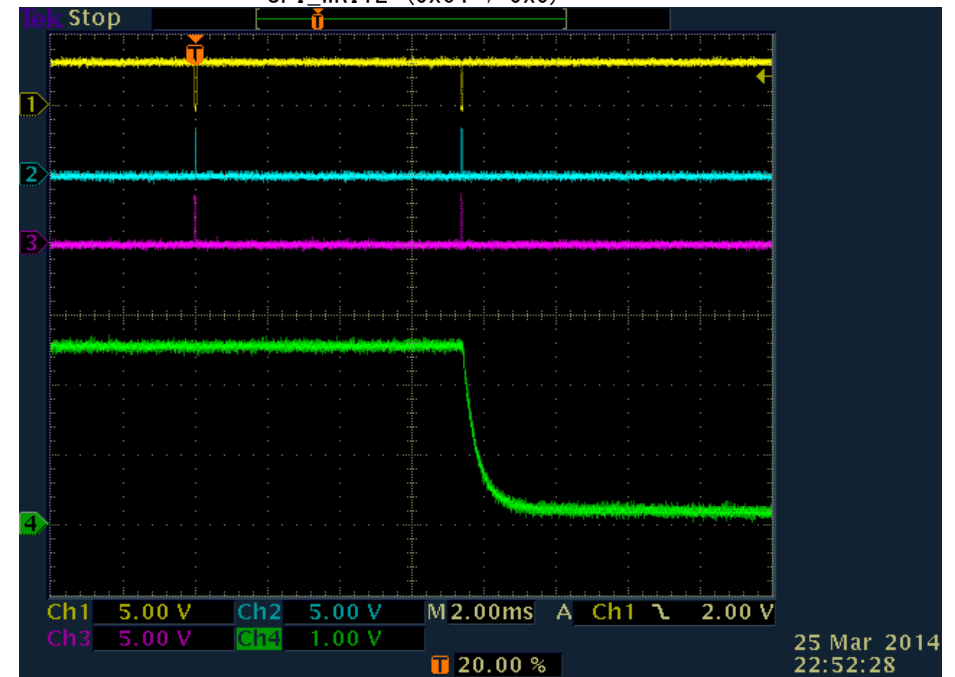
AMUX -> DIAG_OUT waveform with 1k-ohm series resistor (BENCH + TIGER board + GUI software)

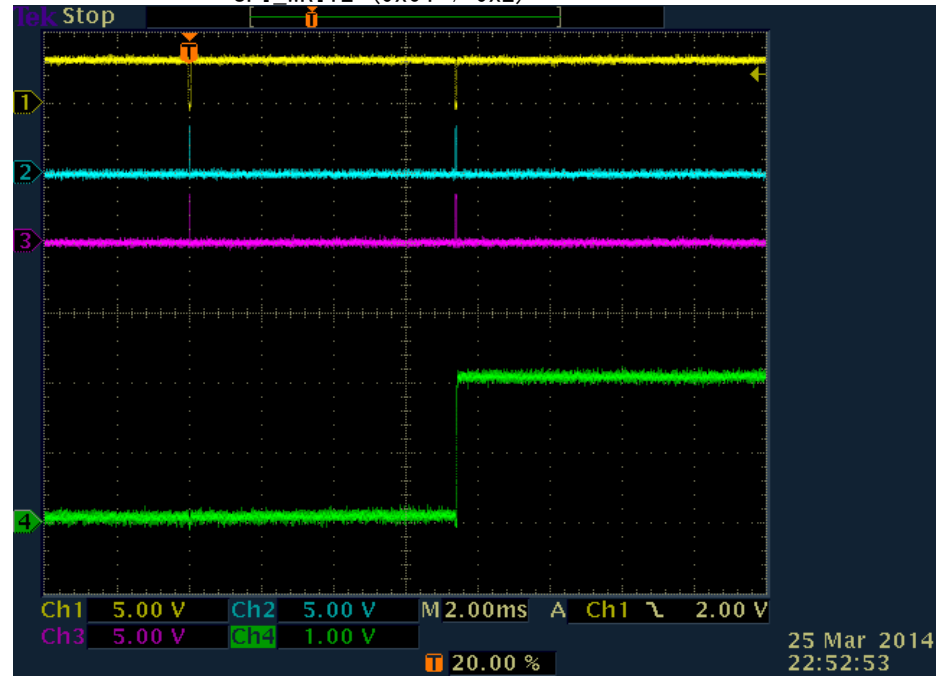A.1 VDD5   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
                  SPI_WRITE (0x64 ; 0x1)   SPI_WRITE (0x64 ; 0x0)
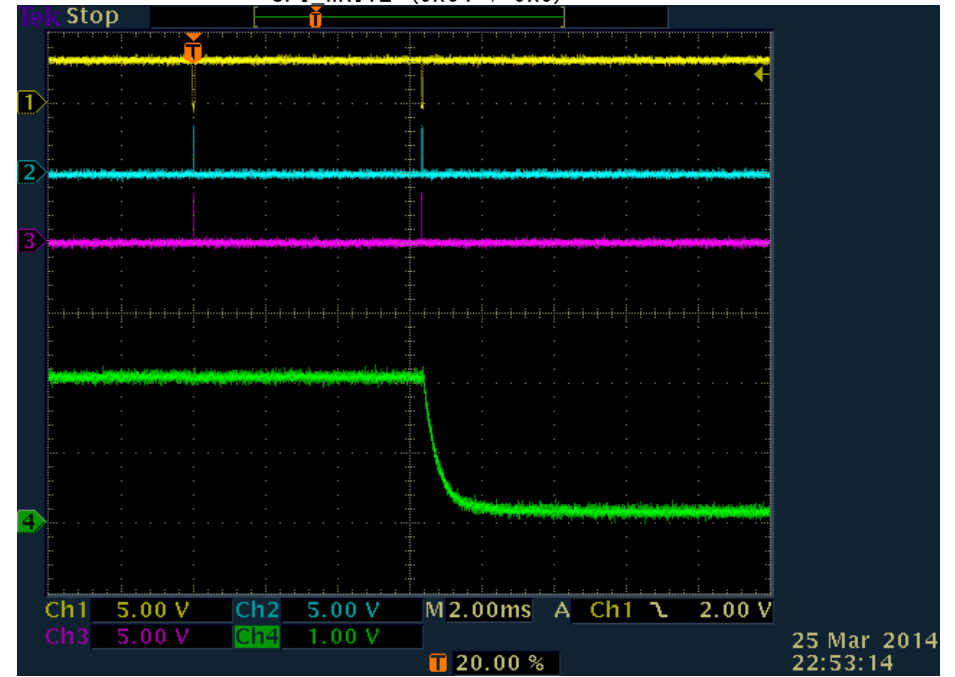

AMUX -> DIAG_OUT waveform with 1k-ohm series resistor (BENCH + TIGER board + GUI software)

A. 2 VDD6

SPI_WRITE (0x66 ; 0xa2)
SPI_WRITE (0x64 ; 0x2)

SPI_WRITE (0x66 ; 0xa2)
SPI_WRITE (0x64 ; 0x0)

AMUX -> DIAG_OUT waveform with 1k-ohm series resistor (BENCH + TIGER board + GUI software)
A.3 VCP   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
                 SPI_WRITE (0x64 ; 0x4)   SPI_WRITE (0x64 ; 0x0)

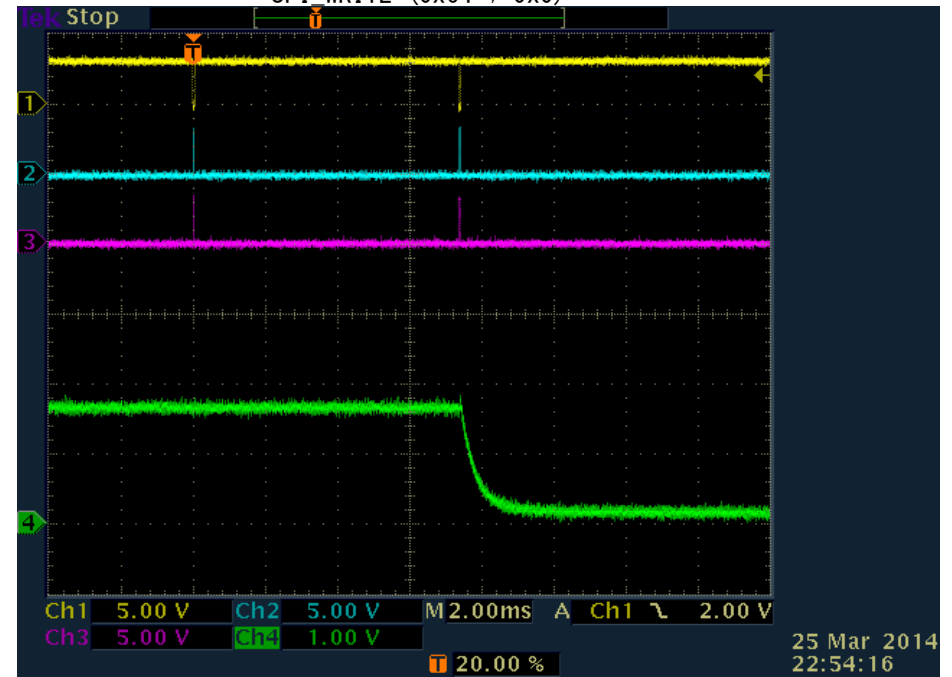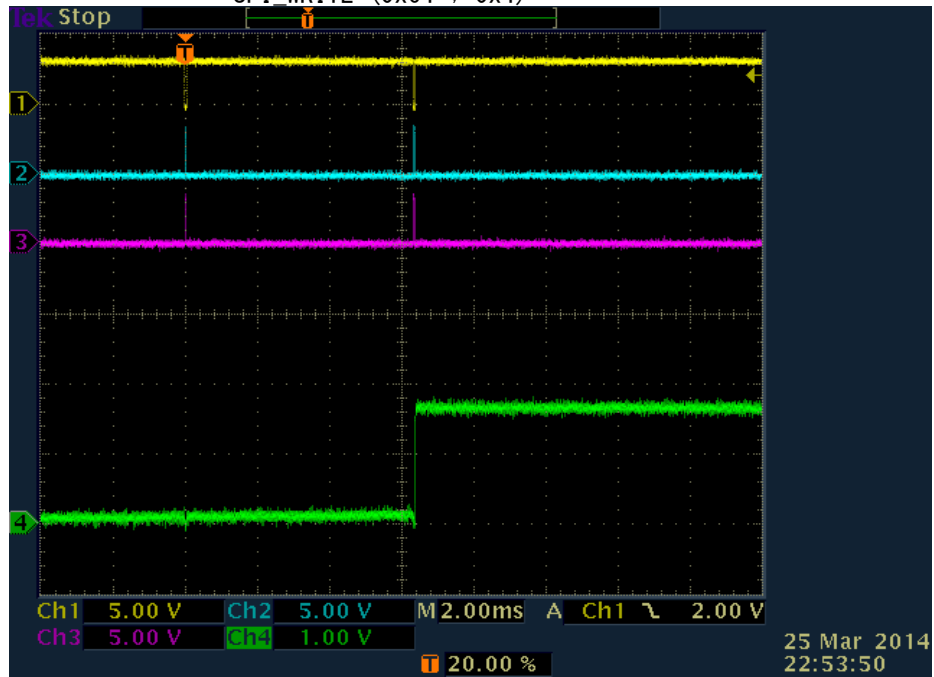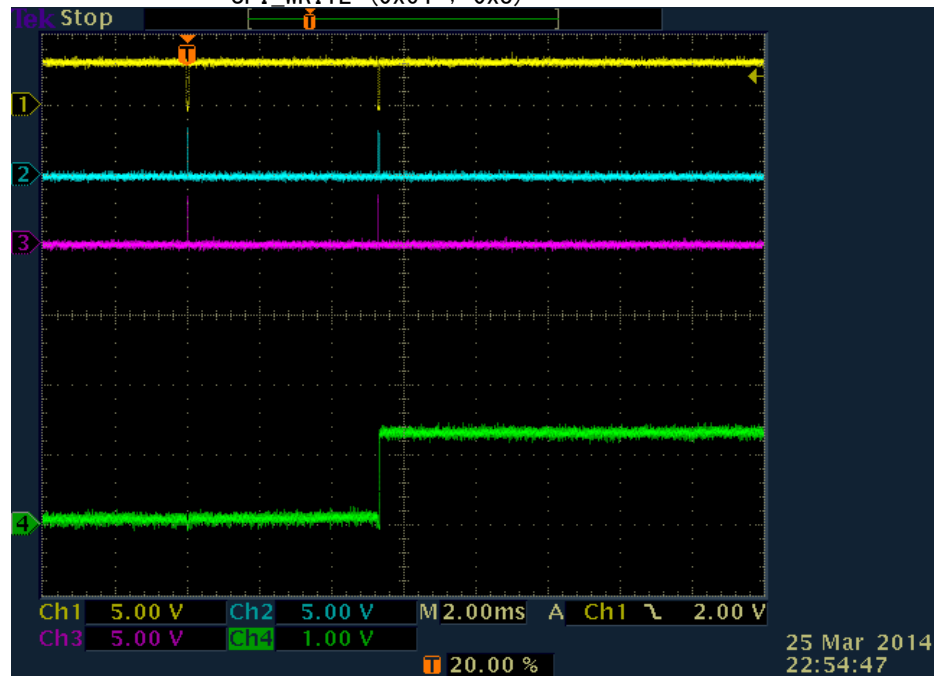AMUX -> DIAG_OUT waveform with 1k-ohm series resistor (BENCH + TIGER board + GUI software)
A.4 VSOUT1   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
                    SPI_WRITE (0x64 ; 0x8)   SPI_WRITE (0x64 ; 0x0)

AMUX -> DIAG_OUT waveform with 1k-ohm series resistor (BENCH + TIGER board + GUI software)
A.5 VBAT_SAFING SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
                 SPI_WRITE (0x64 ; 0x10)   SPI_WRITE (0x64 ; 0x0)

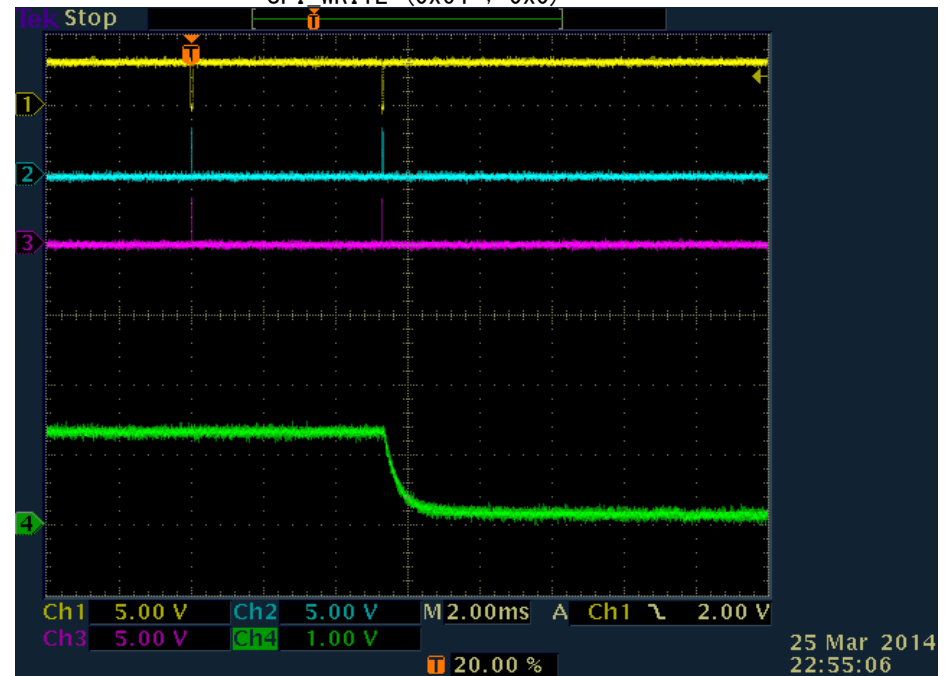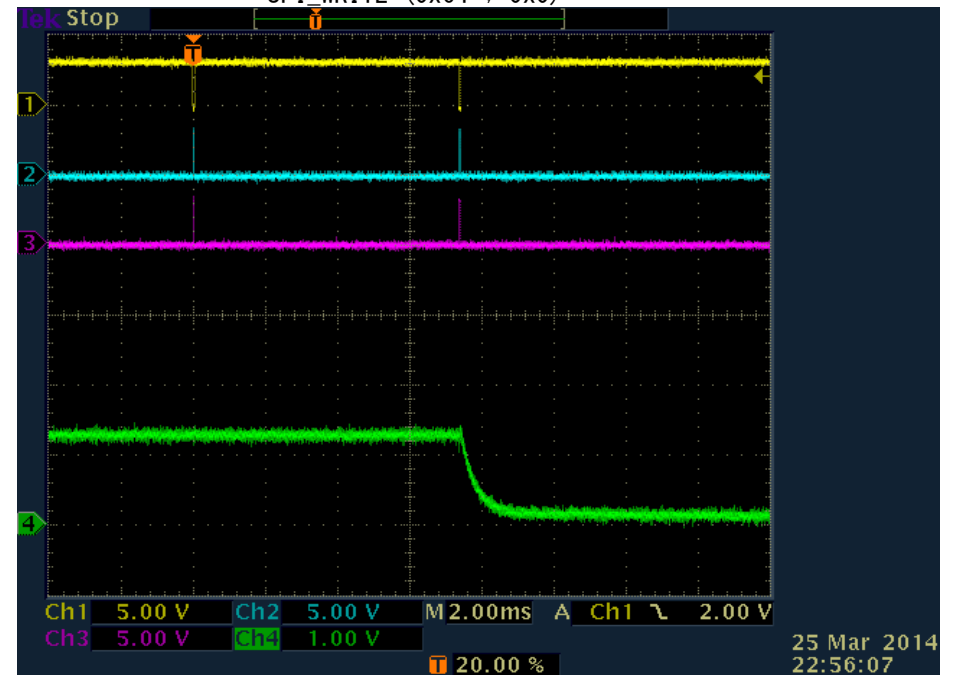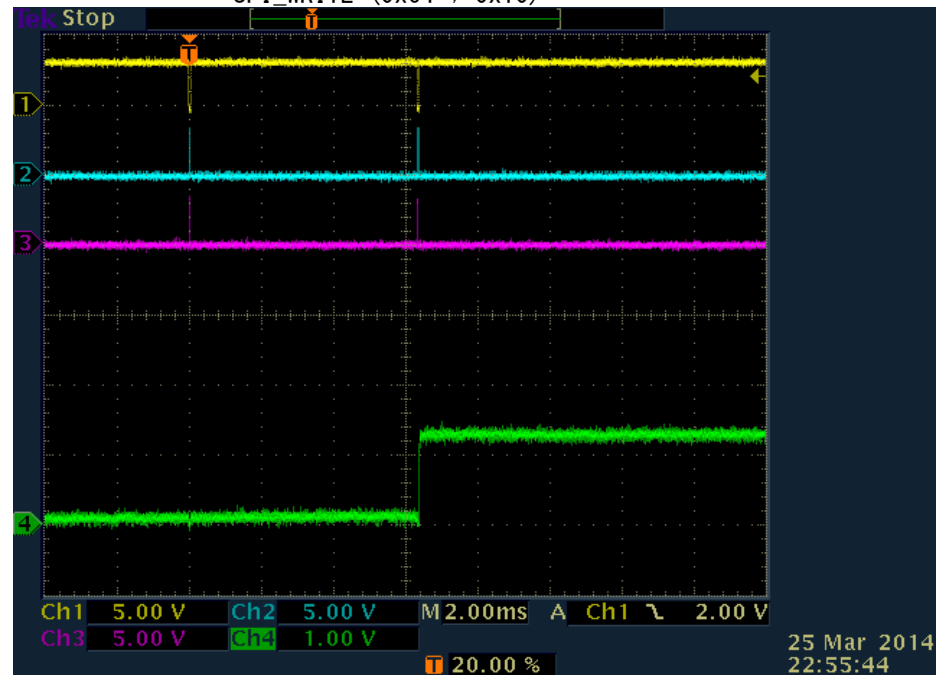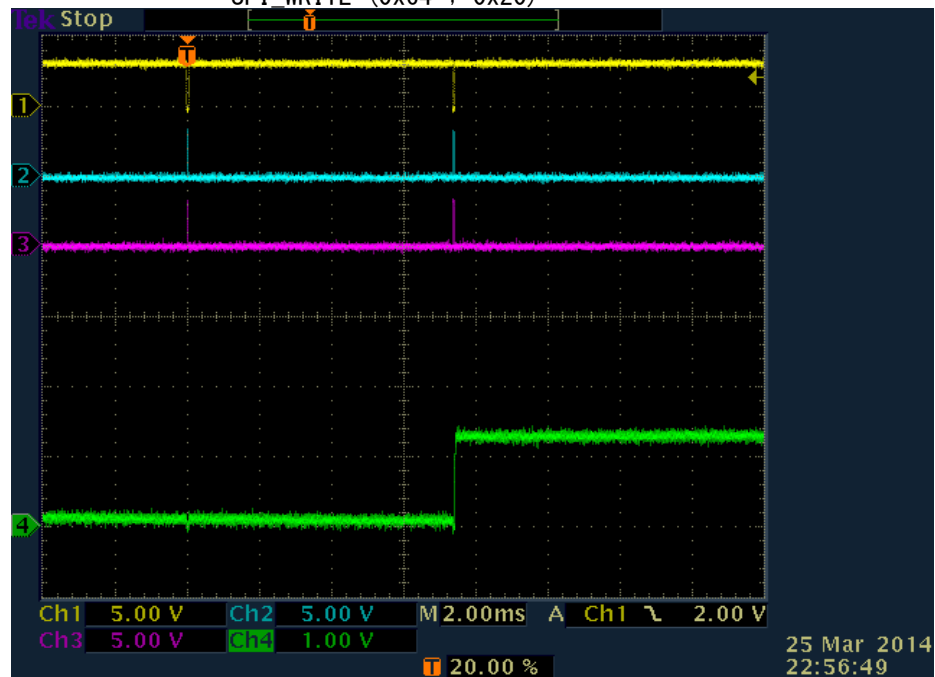AMUX -> DIAG_OUT waveform with 1k-ohm series resistor (BENCH + TIGER board + GUI software)
A.6 VBATP   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
                   SPI_WRITE (0x64 ; 0x20)   SPI_WRITE (0x64 ; 0x0)

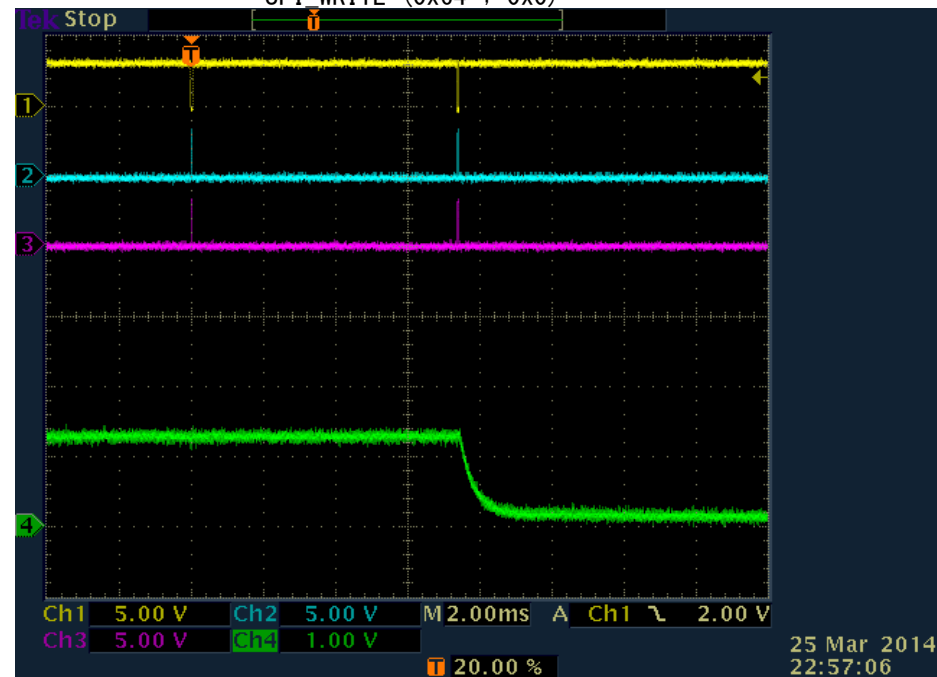AMUX -> DIAG_OUT waveform with 1k-ohm series resistor (BENCH + TIGER board + GUI software)
A.7 MAIN_BG   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
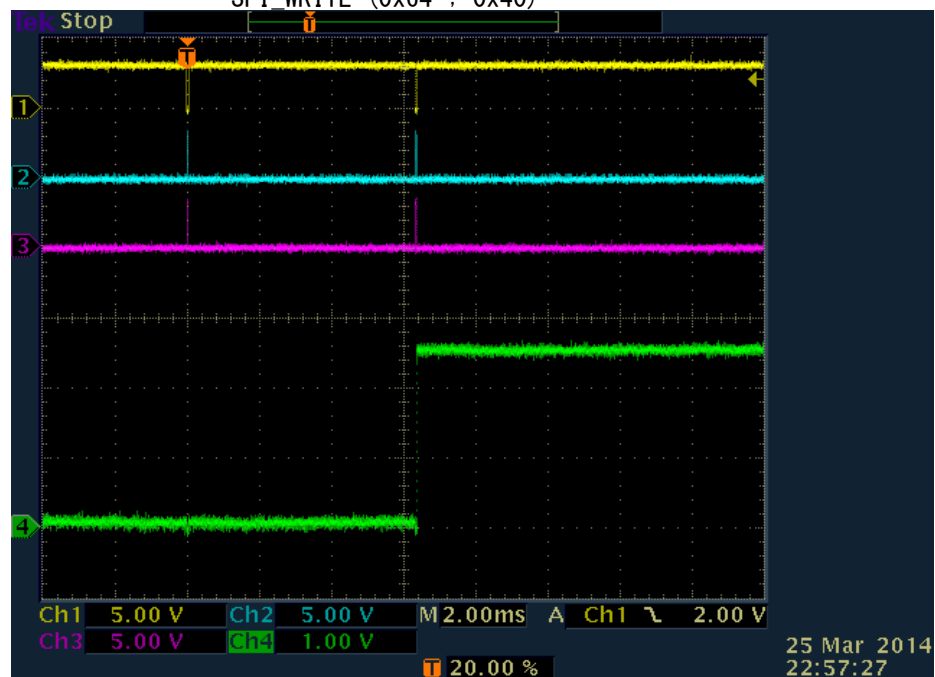   SPI_WRITE (0x64 ; 0x40)   SPI_WRITE (0x64 ; 0x0)

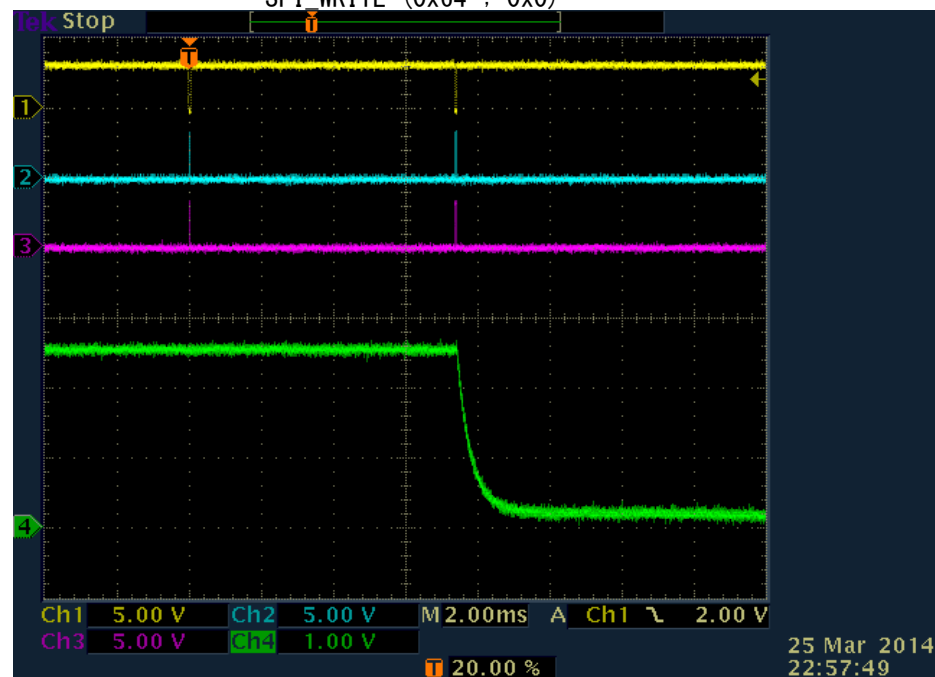AMUX -> DIAG_OUT waveform with 1k-ohm series resistor (BENCH + TIGER board + GUI software)
A.8 VMON_BG   SPI_WRITE (0x66 ; 0xa2)   SPI_WRITE (0x66 ; 0xa2)
                 SPI_WRITE (0x64 ; 0x80)   SPI_WRITE (0x64 ; 0x0)