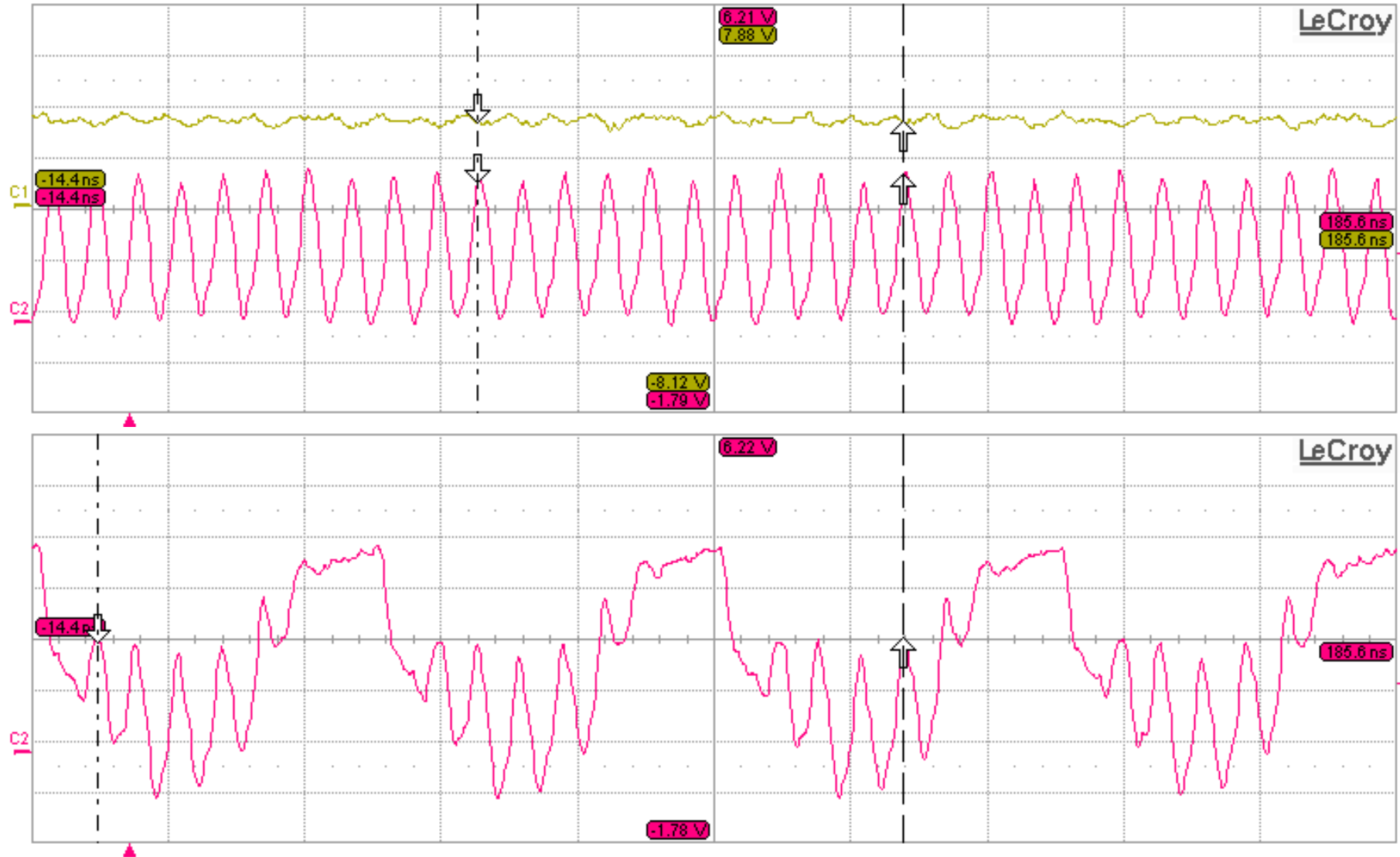# Updates

1. Tested the transaction with repetitive start

2. Tested the configuration files to give commands:

    1. "RD" command need NACK at the end instead of an ACK

    2. A waiting time is necessary after each command

    3. Test all types of transactions

3. Convert the temperature sensor output into degrees

Last update

Clock out @160MHz

Output line (DOUT0)

$2^{OUTPUTMODE}$ = number of serializer lines activated

OUTPUTMODE = controlled with jumpers on the proximity board

Registers Configuration

Name	Bits	Reset	Description
SEL_PHASE	7-4	0	320 MHz Clock phase adjustment to adjust internal and external 40 MHz clocks
EN_SYNC	3	0	Enable synchronization of multiple MIMOSIS1 through SYNC_IN_P and SYNC_IN_N SLVS pads
SEL_START	2-1	0	Start mode selection: Automatic: 0 (by default after PLL locked) Software: 1 (by sending 03 to INSTR CMDID see section 2 page 10) External: 2 or 3 (via START_IN_P and START_IN N SLVS pad)
SEL_CLOCK	0	0	Clock selection: Main @ 40 MHz (CLK_P and CLK_N SLVS pad): 0 Rescue @ 320 MHz[4] (CLKRESCUE_P CLKRESCUE_N SLVS pad): 1 ←

?

Name	Bits	Reset	Description
Not Used	7-6	0	
EN_START_SYNC_TERM	5	0	Enable START_IN and SYNC_IN SLVS 100 Ω termination
FILTER_VAL	4-1	0	Filter glitches on PLL lock flag of N times main clock period (25 ns)
DIS_LOCK_GATING	0	0	Disable generated clocks gating with filtered PLL lock flag ←

Name	Bits	Reset	Description
EN_CLKRESCUE_TERM	7	0	Enable rescue clock (320 MHz) SLVS 100 Ω termination
EN_CLKSER	6	0	Enable output clock serializer (160 MHz) test option ←
EN_CUSTOM_FLV	5	0	Enable custom fill level for elastic buffer test option
EN_CUSTOM_BW	4	0	Enable custom bandwidth for frame generator test option
EN_PATTERN	3	0	Enable pattern for serializer test mode
EN_PULSEINJ	2	0	Enable external pulse injection test option
EN_PIXELMASK	1	1	Enable Pixel Masked (1) or Pulsed (0) mode for pixel array
EN_MFE	0	0	Enable Multi Frame Emulator test mode

Questions and Problems

1. Random responses from the chip when the registers are read (every command is checked with the oscilloscope):

Questions and Problems

1. Random responses from the chip when the registers are read (every command is checked with the oscilloscope):

    1. Problems in the writing stage $\longrightarrow$ We have no control over what we activate and deactivate, but we can have control by reading the register every time after writing it.

Questions and Problems

1. Random responses from the chip when the registers are read (every command is checked with the oscilloscope):

    1. Problems in the writing stage $\longrightarrow$ We have no control over what we activate and deactivate, but we can have control by reading the register every time after writing it.

    2. Problems in the reading stage $\longrightarrow$ We don't have robust control over what we have written, but it doesn't significantly affect the use of the chip.

Questions and Problems

1. Random responses from the chip when the registers are read (every command is checked with the oscilloscope):

    1. Problems in the writing stage ⟶ We have no control over what we activate and deactivate, but we can have control by reading the register every time after writing it.

    2. Problems in the reading stage ⟶ We don't have robust control over what we have written, but it doesn't significantly affect the use of the chip.

    3. Problems in the writing/reading stages ⟶ We have no control over what we are doing.

Questions and Problems

1. Random responses from the chip when the registers are read (every command is checked with the oscilloscope):
   1. Problems in the writing stage $\longrightarrow$ We have no control over what we activate and deactivate, but we can have control by reading the register every time after writing it.
   2. Problems in the reading stage $\longrightarrow$ We don't have robust control over what we have written, but it doesn't significantly affect the use of the chip.
   3. Problems in the writing/reading stages $\longrightarrow$ We have no control over what we are doing.

2. Meaning of the signals on the output lines

Questions and Problems

1. Random responses from the chip when the registers are read (every command is checked with the oscilloscope):

    1. Problems in the writing stage $\longrightarrow$ We have no control over what we activate and deactivate, but we can have control by reading the register every time after writing it.

    2. Problems in the reading stage $\longrightarrow$ We don't have robust control over what we have written, but it doesn't significantly affect the use of the chip.

    3. Problems in the writing/reading stages $\longrightarrow$ We have no control over what we are doing.

2. Meaning of the signals on the output lines

3. What is the meaning of the start, and how can I control it?