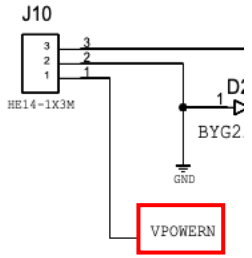
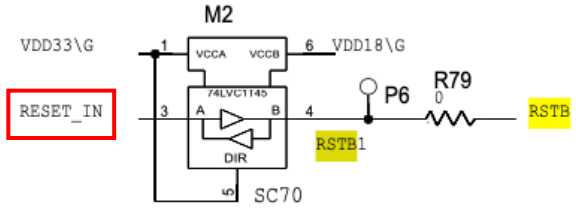
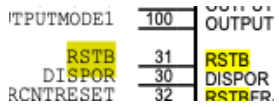
What we have changed to initiate the communications:

1. 

-6 … 0 V → Set to 0 V

2. RESET_IN, SYNC_IN, START_IN can be controlled through SWITCHES (1, 2, 3) [controllable also with 3 bit (24, 8, 16) in the register 0]

3. 

… 

→ To communicate with the chip, the RESET_IN should be set to 1 → Change into the firmware to set 0 the RESET_IN when we want to communicate

Tests: Commands

Name	Value
INSTR	1
ADD_MSB	2
ADD_LSB	3
WR	4
RD	5
WR_IND	6
RD_IND	7
WR_OFF	8
RD_OFF	9

→ Global Reset ✓ (VERY tested!!!)

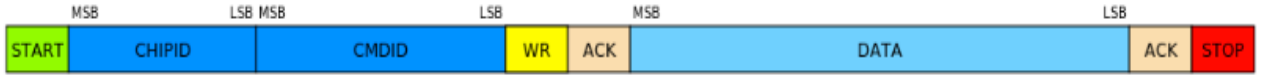
WR, RD → TESTED the Single and Multi byte transactions (next slides)

WR_IND: NOT TESTED

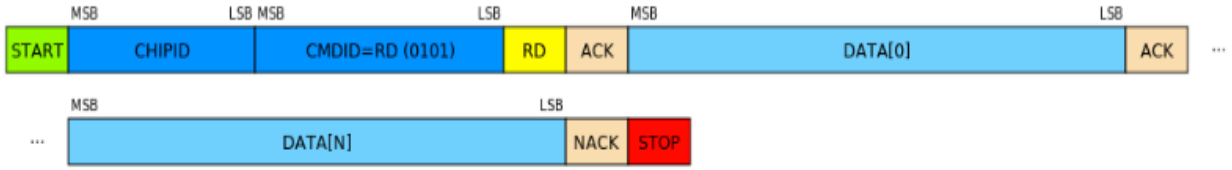
RD_IND: ✓ ✗ → Work with General Configuration registers
DON'T work with Readout Test Configuration registers → It is necessary to understand how the offset is managed for the various registers (see Table 6).

EACH COMMUNICATION IS DISPLAYED ON THE OSCILLOSCOPE TO CHECK THE COMMUNICATION PROTOCOL

Single and Multi byte transactions

1. 

Single byte transaction

2. 

Multi-byte transaction
Non-incremental mode

NOT TESTED

3. 

Multi-byte transaction
Non-incremental mode
With repetitive start

UPDATED today
Not yet tested

4. 

Multi-byte transaction
Incremental mode

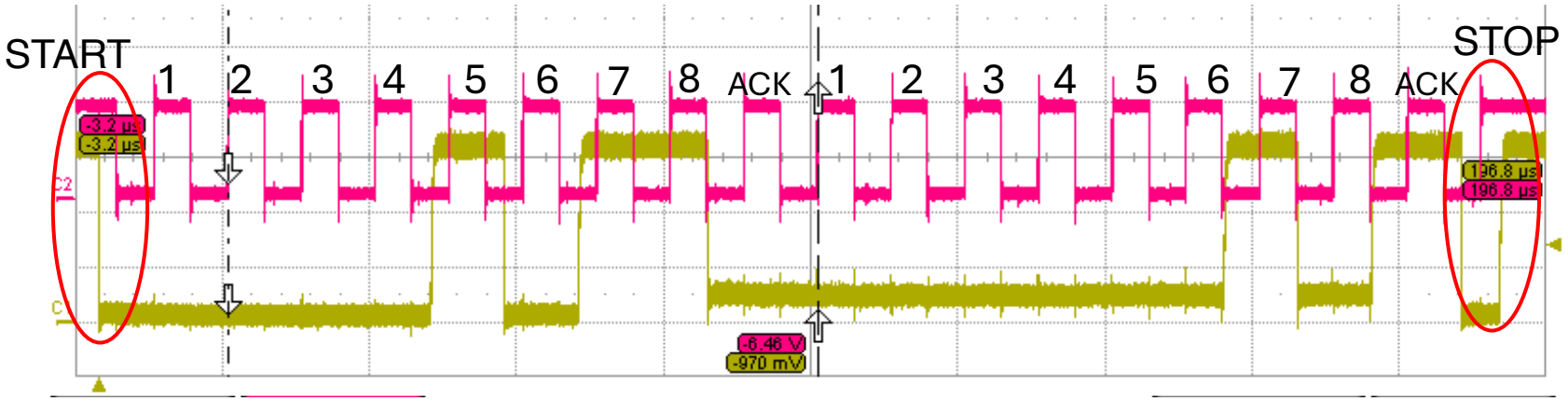
Necessary to place a jumper on J22 (between 3 and 4) to activate the incremental mode (ENINCR).

5. 

Multi-byte transaction
Incremental mode
With repetitive start

UPDATED today
Not yet tested

Communication Protocol

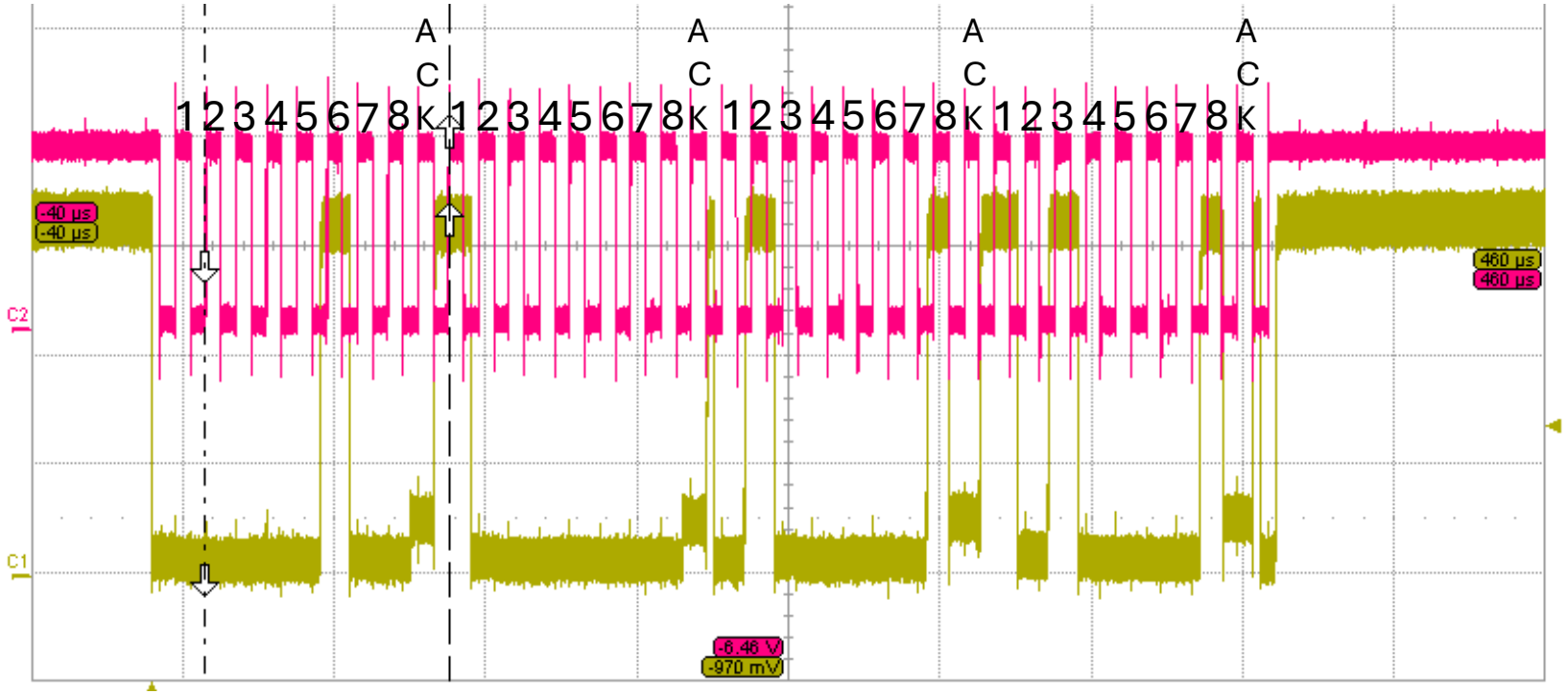


READ of the General Configuration/RUNMODE register

0 0 0 0 1 0 1 1

Chip ID (= 0)
CMDID (= 5)
Write/Read bit (= 1)

0 0 0 0 0 0 1 0
= 02
(default value)



WRITE “a1” in incremental mode the register 8041 (Read Out Test Configuration/PATTERN1 register)

0 0 0 0 0 1 0 0

Chip ID (= 0)
CMDID (= 2)
Write/Read bit (= 0)

1 0 0 0 0 0 0 0
= 80
(MSB)

0 1 0 0 0 0 0 1
= 41
(LSB)

1 0 1 0 0 0 0 1
= a1

Tests: Registers

Function	Bank number	
General Configuration	1	→ Read the Reset value of all the registers
DAC	2	NOT TESTED
Sequencer Configuration	3	NOT TESTED
Pixel Control	4	NOT TESTED
Monitoring	7	→ MON_PAD ✓ →
Multi-Frame Emulation	8	NOT TESTED
Analogue Pixel Selection	9	NOT TESTED
Readout Configuration	10	→ Read/Write on the registers (IDs: 8040-8053)

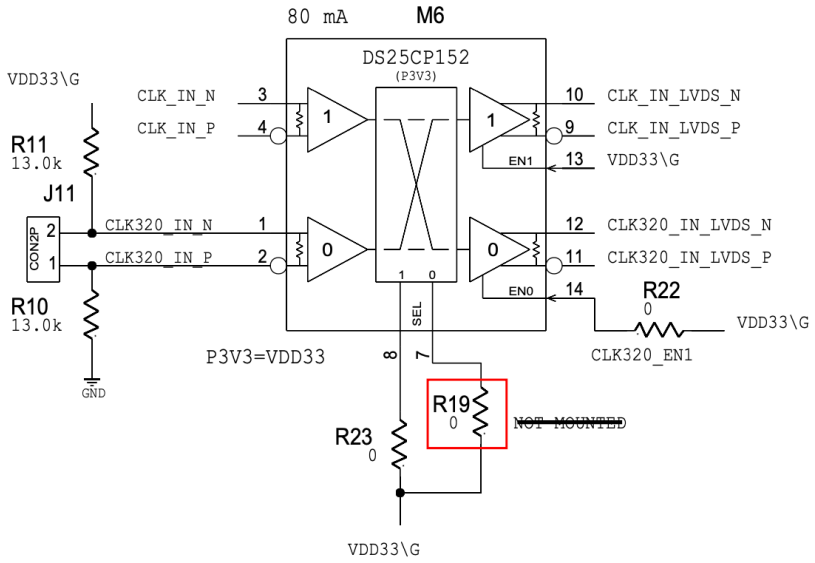
Name	Bits	Reset
Not Used	7-6	0
DISPOR	5	-
ENINCR	4	-
SYNC_IN	3	-
START_IN	2	-
OUTPUTMODE	1-0	-

In general, sometimes, regardless of the register we are reading, the chip responds with incorrect data. Everything remains consistent with the communication protocol, so it seems to be an issue with the chip itself rather than the way we are communicating.

Problems with even registers

Register ID	Written word	Read word	
8040	a0	40	✗
8041	a1	a1	✓
8042	a2	42	✗
8043	a3	a3	✓

Tests: Clock-in/out



External clock (R19 mounted)

↓

Clock Rescue mode (not used the PLL)

↓

New firmware with 320 MHz clock

We still can't see the clock out.

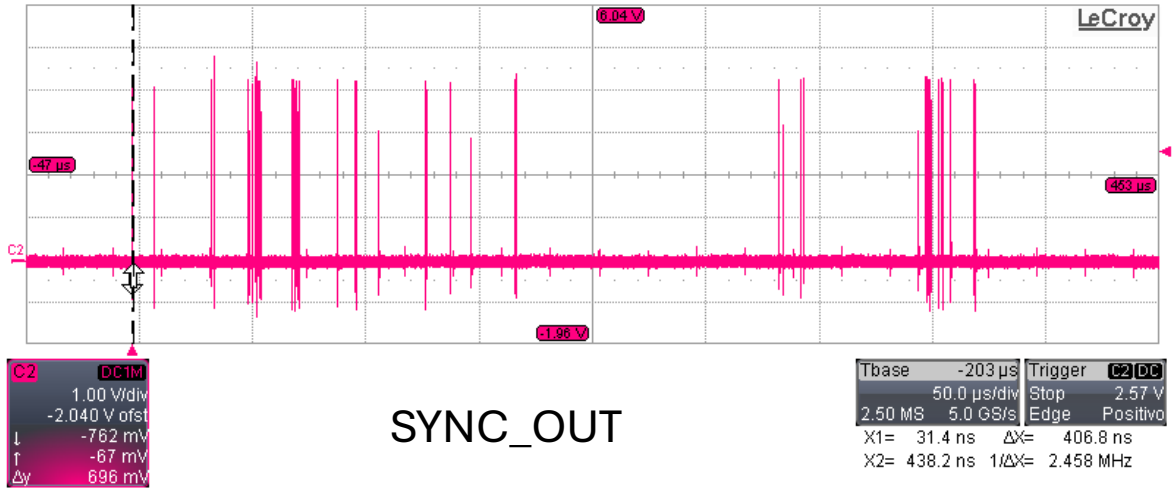
CLKGEN1	0027	00
CLKGEN2	0028	00
PLL	0029	11
PLLLOCK	002A	38

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

Next steps…

- Tests of the new firmware with the possibility of multiple START
- Tests of the configuration files as input
- Try to understand the meaning of some signals: SYNC_OUT, TMON, …
- Understand the actual configuration of the board and the significant bit to change to see the clock-out



SYNC_OUT