# Stratasys One-Wire EEPROM Communication Analysis

EEPROM Device Family Code: 0xC0

Protocol: Based on Dallas/Maxim 1-Wire, with custom extensions

# Known Commands and Behavior

| Command | Code | Description |
|---|---|---|
| Match ROM | 0x55 | Select specific slave via 64-bit ID |
| Search ROM | 0xF0 | Standard enumeration command |
| Read Memory data + 2-byte CRC | 0xF0 + page | Reads memory, response: 2-byte header + 32 bytes |
| Read Scratchpad | 0xAA + offset | Reads scratchpad, reused for multiple address ranges |
| Write Scratchpad | 0x0F + offset | Starts a write operation, followed by 32 bytes from master |
| Commit Scratchpad | 0xA5 + page | Writes scratchpad data to memory page |
| Ready/Status Signal | (e.g., 0xAA) | Slave signals readiness between steps |
| Reset | — | Standard 1-Wire bus reset |

# Memory Architecture (Inferred)

Page size: 32 bytes

Pages observed: at least 0x00 through 0xA0 (implies 5+ pages = 160 bytes)

Read format: Always 2 + 32 + 2 bytes

Write flow:

0x0F + offset — Write Scratchpad

Master sends 32 bytes

0xA5 + page — Commit to EEPROM

Slave acknowledges and may auto-send memory dump for verification

# General Communication Patterns

Memory Read

(M)aster: Reset → 0x55 + ID → 0xF0 + page

(S)lave: 2 + 32 + 2 bytes

Memory Write

M: Reset → 0x55 + ID

M: 0x0F + 0x80 (scratchpad offset)

S: 2 bytes

M: Send 32 bytes

S: 2 bytes (CRC?)

M: Reset → 0x55 + ID

M: 0xA5 + page

S: 1 byte ack → pause → 1 byte done

S: 32 bytes (written data) + 2 CRC

S: [Sometimes] 32 more + 2 bytes (next page?)

Custom/Unexpected Behavior

- 0xAA used both for scratchpad and memory reads.
- Sometimes returns 2 + 4 + 2 bytes instead of full page — likely special region or register space.
- Responses like 0xAA used for signaling between phases.
- 0x80 always used as scratchpad offset. (Scratchpad confirmed to be here)

# Observed Waveform Behavior

Section 1

Standard search, match, and multiple memory reads.

Verifies format: 2 + 32 + 2 per memory read.

Scratchpad reads at offset 0xE0, 0x20.

Section 2

Multiple reads from memory pages via 0xAA

First write cycle: 0x0F + 0x80 → 32 bytes → 0xA5 + 0x00

Long pause before final dump: likely EEPROM commit time

Slave echoes written memory

Section 3

Another write to page 0x01

Verifies same write process

Auto-dump of two memory pages after write

Read confirms successful EEPROM write

Section 4

Reads from higher address 0xA0 succeed

Second read from 0x00 returns full data

Final read from 0x00 returns only 4 bytes, a deviation:

Possibly status/config/meta region

May indicate partial scratchpad or non-memory space

# Hypotheses & Key Findings

EEPROM has custom ROM function codes and enhanced memory protocol.

0xA5 is custom write-commit command, with simplified args (no E/S byte).

Chip may automatically verify memory by echoing back written data.

2 + 4 + 2-byte response could mean:

- Device transitioned into a status/config space

- Or truncated data due to partial/invalid memory

# Potential **Next Steps**

Share raw hex of 2 + 4 + 2-byte segment

Try accessing other offset ranges with 0xAA (e.g., 0x10, 0x08)

Probe for additional hidden registers by writing to or reading from small offsets

Confirm max memory boundary by writing past 0xA0