ZeroII Vector Antenna Analyzer

Short communication interfaces description.

The ZeroII in measurement mode supports communication with the host controller via SPI, I2C, UART. I2C interface use 7-bit slave address 0x5B. The firmware is updated via the USB interface using the RigExpert Flash Tool utility.

Commands supported by SPI, I2C, UART interfaces:

ZEROII_GET_STATUS (0x5A) – ask for the ZeroII status. (Except SPI - you can read the status after setting CS low without prompting).

Possible answers:

ZEROII_STATUS_BUSY_USB (0x01) – the device is busy with a task for the USB interface;

ZEROII_STATUS_BUSY_SPI (0x02) – the device is busy with a task for the SPI interface;

ZEROII_STATUS_BUSY_I2C (0x03) – the device is busy with a task for the I2C interface;

ZEROII_STATUS_BUSY_UART (0x04) – the device is busy with a task for the UART interface;

ZEROII_STATUS_IDLE (0x05) – the device is idle, ready for new tasks;

ZEROII_STATUS_READY (0x06) – measurement results are ready;

ZEROII_STATUS_ERROR (0x07) – an error occurred while executing the task;

Example request:

[(uint8_t)CMD, (uint8_t)CRC ***1**, (uint8_t)INV_CRC ***1**]

0x5A, 0x81, 0x7E

Response (immediate):

[(uint8_t)Status = *IDLE*, (uint8_t)CRC ***1**, (uint8_t)INV_CRC ***1**]

0x05, 0x1B, 0xE4

ZEROII_SET_SYSTEM_Z0 (0xF2) – set the system impedance value (Z0). The default is 50 ohms.

Example request:

[(uint8_t)CMD, (`uint32_t`)Z0=50000 mOhm, (uint8_t)CRC ***1**, (uint8_t)INV_CRC ***1**]

0xF2, 0x50, 0xC3, 0x00, 0x00, 0x01, 0xFE

Response: NONE

ZEROII_GET_SYSTEM_Z0 (0xC4) – get the system impedance value (Z0).

Example request:

[(uint8_t)CMD, (uint8_t)CRC ***1**, (uint8_t)INV_CRC ***1**]

0xC4, 0x52, 0xAD

Response (immediate):

[(`uint32_t`)Z0=50000 mOhm, (uint8_t)CRC ***2**, (uint8_t)INV_CRC ***1**]

0x50, 0xC3, 0x00, 0x00, 0xCC, 0x33

ZEROII_SET_FQ_GET_RX (0x6D) – set the measurement frequency, get the results as R (resistance), X (reactance).

Example request:

[(uint8_t)CMD, (uint32_t)FQ=14 720 000, (uint8_t)CRC *[1], (uint8_t)INV_CRC *[1]]

0x6D, 0x00, 0x9C, 0xE0, 0x00, 0x48, 0xB7

Response (delayed *[3]):

[(float)R=50.1416, (float)X=0.314159, (uint8_t)CRC *[2], (uint8_t)INV_CRC *[1]]

0xFD, 0x90, 0x48, 0x42, 0x7A, 0xD9, 0xA0, 0x3E, 0x88, 0x77

ZEROII_SET_FQ_GET_RXSWRRL (0xA3) – set measurement frequency, get results as R (resistance), X (reactance), SWR (standing wave ratio), RL (return loss).

Example request:

[(uint8_t)CMD, (uint32_t)FQ=14 720 000, (uint8_t)CRC *[1], (uint8_t)INV_CRC *[1]]

0xA3, 0x00, 0x9C, 0xE0, 0x00, 0x45, 0xBA

Response (delayed *[3]):

[(float)R=50.1416, (float)X=0.314159, (float)SWR=1.03742, (float)RL=34.5816, (uint8_t)CRC *[2], (uint8_t)INV_CRC *[1]]

0xFD, 0x90, 0x48, 0x42, 0x7A, 0xD9, 0xA0, 0x3E, 0x2E, 0xCA, 0x84, 0x3F, 0x8F, 0x53, 0x0A, 0x42, 0x38, 0xC7

ZEROII_GET_RX_DATA (0x7C) – repeat the measurement at the previous frequency, get the results as R (resistance), X (reactance).

Example request:

[(uint8_t)CMD, (uint8_t)CRC *[1], (uint8_t)INV_CRC *[1]]

0x7C, 0x73, 0x8C

Response (delayed *[3]):

[(float)R=50.1416, (float)X=0.314159, (uint8_t)CRC *[2], (uint8_t)INV_CRC *[1]]

0xFD, 0x90, 0x48, 0x42, 0x7A, 0xD9, 0xA0, 0x3E, 0x88, 0x77

ZEROII_GET_RX_SWR_RL (0x9A) – repeat the measurement at the previous frequency, get the results as R (resistance), X (reactance), SWR (standing wave ratio), RL (return loss).

Example request:

[(uint8_t)CMD, (uint8_t)CRC *[1], (uint8_t)INV_CRC *[1]]

0x9A, 0xCF, 0x30

Response (delayed *[3]):

[(float)R=50.1416, (float)X=0.314159, (float)SWR=1.03742, (float)RL=34.5816, (uint8_t)CRC *[2], (uint8_t)INV_CRC *[1]]

0xFD, 0x90, 0x48, 0x42, 0x7A, 0xD9, 0xA0, 0x3E, 0x2E, 0xCA, 0x84, 0x3F, 0x8F, 0x53, 0x0A, 0x42, 0x38, 0xC7

ZEROII_GET_FW_VERSION (0xE5) – ask the current firmware version, hardware revision, serial number of the device.

Example request:

[(uint8_t)CMD, (uint8_t)CRC *****[1], (uint8_t)INV_CRC *****[1]]

0xE5, 0xB5, 0x4A

Response (immediate):

[(uint8_t)MAJOR_VER = 1, (uint8_t)MINOR_VER = 1, (uint8_t)HW_REV = 1, (uint32_t)SN = 400107968, (uint8_t)CRC *****[2], (uint8_t)INV_CRC *****[1]]

0x01, 0x01, 0x01, 0xC0, 0x29, 0xD9, 0x17, 0x25, 0xDA

***1** – used for UART only (INV_CRC = CRC ^ 0xFF).

***2** – used for SPI and UART.

*3 – before receiving delayed response you need at least once ask for STATUS and get *ZEROII_STATUS_READY*. Until the measurement is completed, STATUS will correspond to *ZEROII_STATUS_BUSY_x.*