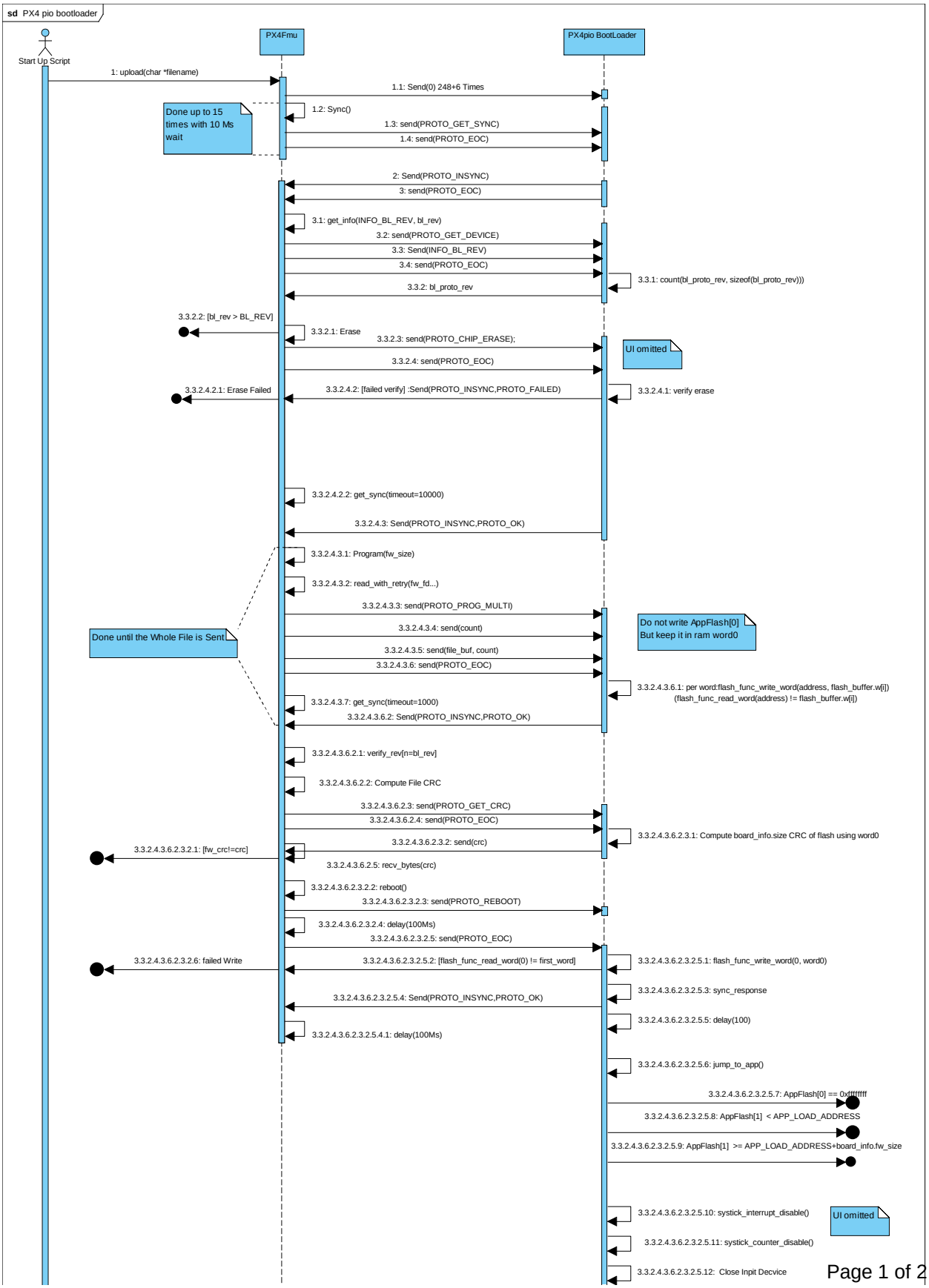
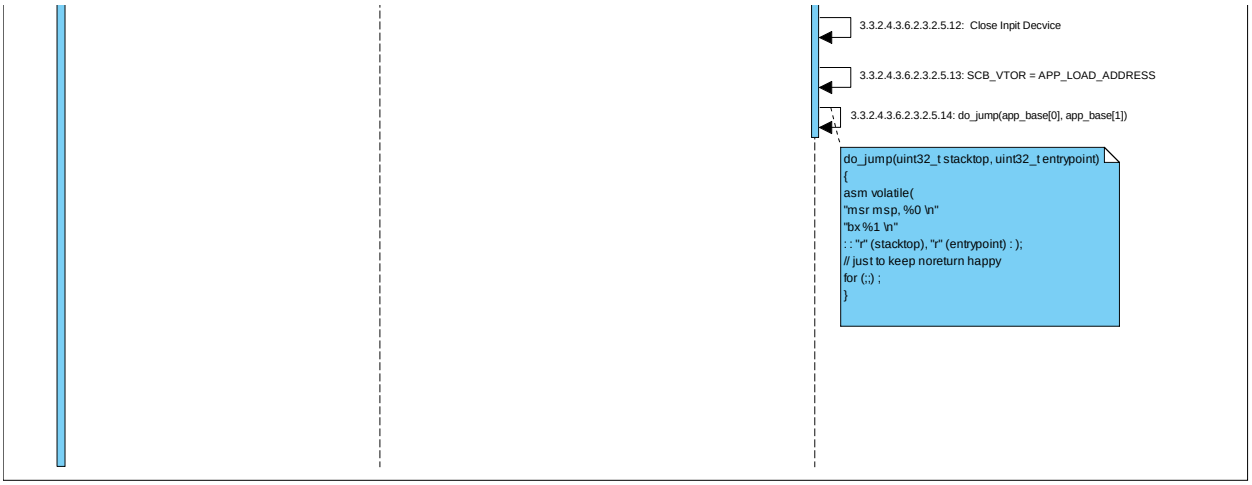

sd PX4 pio bootloader
Start Up Script
PX4Fmu
PX4pio BootLoader
1: upload(char *filename)
1.1: Send(0) 248+6 Times
1.2: Sync()
Done up to 15 times with 10 Ms wait
1.3: send(PROTO_GET_SYNC)
1.4: send(PROTO_EOC)
2: Send(PROTO_INSYNC)
3: send(PROTO_EOC)
3.1: get_info(INFO_BL_REV, bl_rev)
3.2: send(PROTO_GET_DEVICE)
3.3: Send(INFO_BL_REV)
3.4: send(PROTO_EOC)
3.3.1: count(bl_proto_rev, sizeof(bl_proto_rev)))
3.3.2: bl_proto_rev
3.3.2.2: [bl_rev > BL_REV]
3.3.2.1: Erase
3.3.2.3: send(PROTO_CHIP_ERASE);
UI omitted
3.3.2.4: send(PROTO_EOC)
3.3.2.4.2.1: Erase Failed
3.3.2.4.2: [failed verify] :Send(PROTO_INSYNC,PROTO_FAILED)
3.3.2.4.1: verify erase
3.3.2.4.2.2: get_sync(timeout=10000)
3.3.2.4.3: Send(PROTO_INSYNC,PROTO_OK)
3.3.2.4.3.1: Program(fw_size)
3.3.2.4.3.2: read_with_retry(fw_fd...)
3.3.2.4.3.3: send(PROTO_PROG_MULTI)
Do not write AppFlash[0]
But keep it in ram word0
Done until the Whole File is Sent
3.3.2.4.3.4: send(count)
3.3.2.4.3.5: send(file_buf, count)
3.3.2.4.3.6: send(PROTO_EOC)
3.3.2.4.3.6.1: per word:flash_func_write_word(address, flash_buffer.w[i])
(flash_func_read_word(address) != flash_buffer.w[i])
3.3.2.4.3.7: get_sync(timeout=1000)
3.3.2.4.3.6.2: Send(PROTO_INSYNC,PROTO_OK)
3.3.2.4.3.6.2.1: verify_rev[n=bl_rev]
3.3.2.4.3.6.2.2: Compute File CRC
3.3.2.4.3.6.2.3: send(PROTO_GET_CRC)
3.3.2.4.3.6.2.4: send(PROTO_EOC)
3.3.2.4.3.6.2.3.1: Compute board_info.size CRC of flash using word0
3.3.2.4.3.6.2.3.2: send(crc)
3.3.2.4.3.6.2.3.2.1: [fw_crc!=crc]
3.3.2.4.3.6.2.5: recv_bytes(crc)
3.3.2.4.3.6.2.3.2.2: reboot()
3.3.2.4.3.6.2.3.2.3: send(PROTO_REBOOT)
3.3.2.4.3.6.2.3.2.4: delay(100Ms)
3.3.2.4.3.6.2.3.2.5: send(PROTO_EOC)
3.3.2.4.3.6.2.3.2.6: failed Write
3.3.2.4.3.6.2.3.2.5.2: [flash_func_read_word(0) != first_word]
3.3.2.4.3.6.2.3.2.5.1: flash_func_write_word(0, word0)
3.3.2.4.3.6.2.3.2.5.3: sync_response
3.3.2.4.3.6.2.3.2.5.4: Send(PROTO_INSYNC,PROTO_OK)
3.3.2.4.3.6.2.3.2.5.5: delay(100)
3.3.2.4.3.6.2.3.2.5.4.1: delay(100Ms)
3.3.2.4.3.6.2.3.2.5.6: jump_to_app()
3.3.2.4.3.6.2.3.2.5.7: AppFlash[0] == 0xffffffff
3.3.2.4.3.6.2.3.2.5.8: AppFlash[1] < APP_LOAD_ADDRESS
3.3.2.4.3.6.2.3.2.5.9: AppFlash[1] >= APP_LOAD_ADDRESS+board_info.fw_size
3.3.2.4.3.6.2.3.2.5.10: systick_interrupt_disable()
UI omitted
3.3.2.4.3.6.2.3.2.5.11: systick_counter_disable()
3.3.2.4.3.6.2.3.2.5.12: Close Inpit Decvice


3.3.2.4.3.6.2.3.2.5.12: Close Inpit Decvice
3.3.2.4.3.6.2.3.2.5.13: SCB_VTOR = APP_LOAD_ADDRESS
3.3.2.4.3.6.2.3.2.5.14: do_jump(app_base[0], app_base[1])
do_jump(uint32_t stacktop, uint32_t entrypoint)
{
asm volatile(
"msr msp, %0 \n"
"bx %1 \n"
: : "r" (stacktop), "r" (entrypoint) :);
// just to keep noreturn happy
for (;;) ;
}