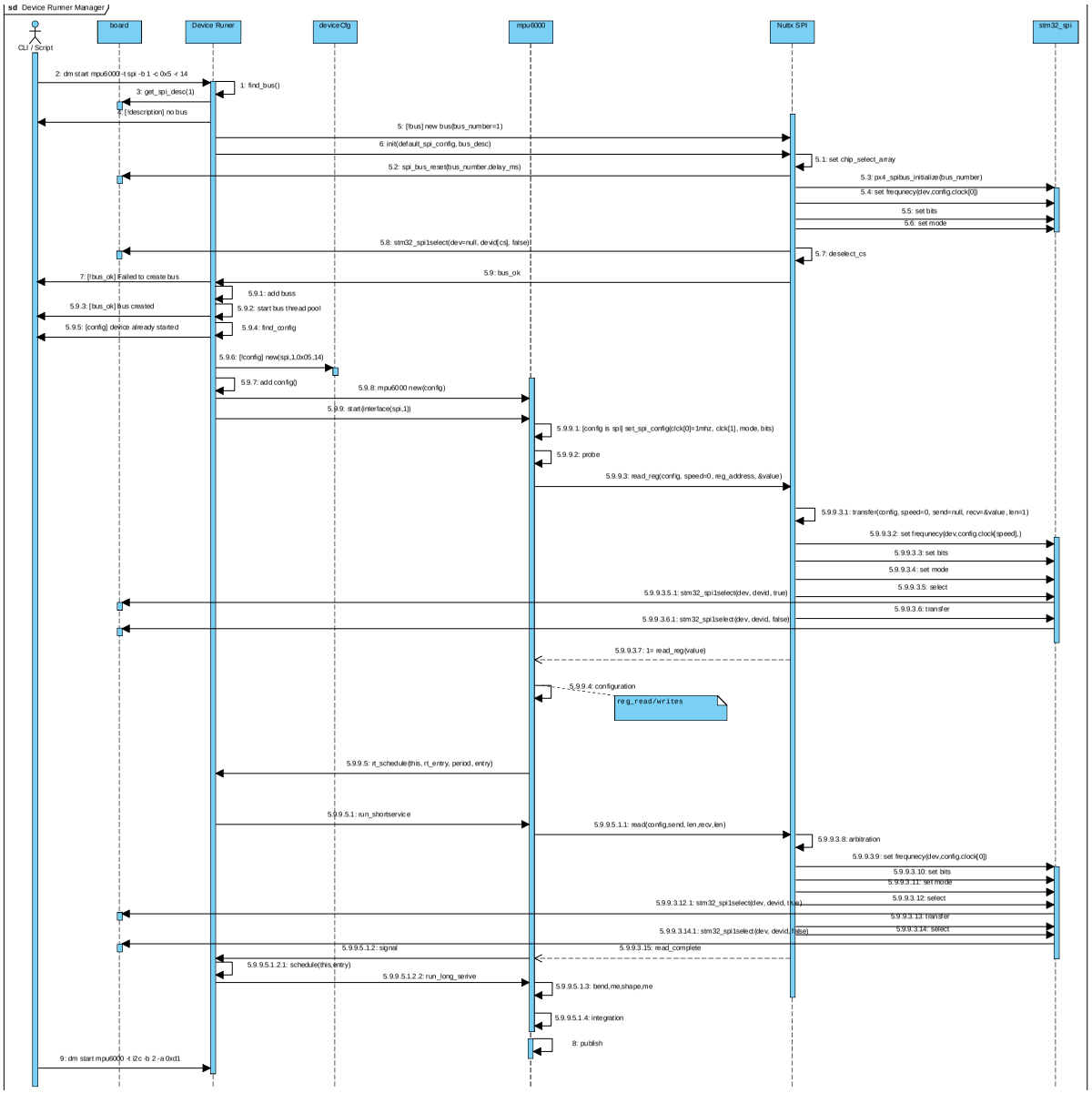
sd Device Runner Manager
CLI / Script
board
Device Runer
deviceCfg
mpu6000
Nuttx SPI
stm32_spi
2: dm start mpu6000 -t spi -b 1 -c 0x5 -r 14
1: find_bus()
3: get_spi_desc(1)
4: [!description] no bus
5: [!bus] new bus(bus_number=1)
6: init(default_spi_config, bus_desc)
5.1: set chip_select_array
5.2: spi_bus_reset(bus_number,delay_ms)
5.3: px4_spibus_initialize(bus_number)
5.4: set frequnecy(dev,config.clock[0])
5.5: set bits
5.6: set mode
5.8: stm32_spi1select(dev=null, devid[cs], false)1
5.7: deselect_cs
5.9: bus_ok
7: [!bus_ok] Failed to create bus
5.9.1: add buss
5.9.3: [bus_ok] Bus created
5.9.2: start bus thread pool
5.9.5: [config] device already started
5.9.4: find_config
5.9.6: [!config] new(spi,1,0x05,14)
5.9.7: add config()
5.9.8: mpu6000 new(config)
5.9.9: start(interface(spi,1))
5.9.9.1: [config is spi] set_spi_config(clck[0]=1mhz, clck[1], mode, bits)
5.9.9.2: probe
5.9.9.3: read_reg(config, speed=0, reg_address, &value)
5.9.9.3.1: transfer(config, speed=0, send=null, recv=&value, len=1)
5.9.9.3.2: set frequnecy(dev,config.clock[speed],)
5.9.9.3.3: set bits
5.9.9.3.4: set mode
5.9.9.3.5: select
5.9.9.3.5.1: stm32_spi1select(dev, devid, true)
5.9.9.3.6: transfer
5.9.9.3.6.1: stm32_spi1select(dev, devid, false)
5.9.9.3.7: 1= read_reg(value)
5.9.9.4: configuration
reg_read/writes
5.9.9.5: rt_schedule(this, rt_entry, period, entry)
5.9.9.5.1: run_shortservice
5.9.9.5.1.1: read(config,send,len,recv,len)
5.9.9.3.8: arbitration
5.9.9.3.9: set frequnecy(dev,config.clock[0])
5.9.9.3.10: set bits
5.9.9.3.11: set mode
5.9.9.3.12: select
5.9.9.3.12.1: stm32_spi1select(dev, devid, true)
5.9.9.3.13: transfer
5.9.9.3.14.1: stm32_spi1select(dev, devid, false)
5.9.9.3.14: select
5.9.9.3.15: read_complete
5.9.9.5.1.2: signal
5.9.9.5.1.2.1: schedule(this,entry)
5.9.9.5.1.2.2: run_long_serive
5.9.9.5.1.3: bend,me,shape,me
5.9.9.5.1.4: integration
8: publish
9: dm start mpu6000 -t i2c -b 2 -a 0xd1