

Error codes	
a)adapter_success	=0;
b)adapter_AF	=1;
c)adapter_BERR	=2;
d)adapter_ARLO	=3;
e)adapter_OVR	=4;
f)adapter_timeout	=5;
g)adapter_other_error	=6;
e)adapter_busy	=7;

Data markers	
a)data_from_host	=18;
b)data_to_host	=19;

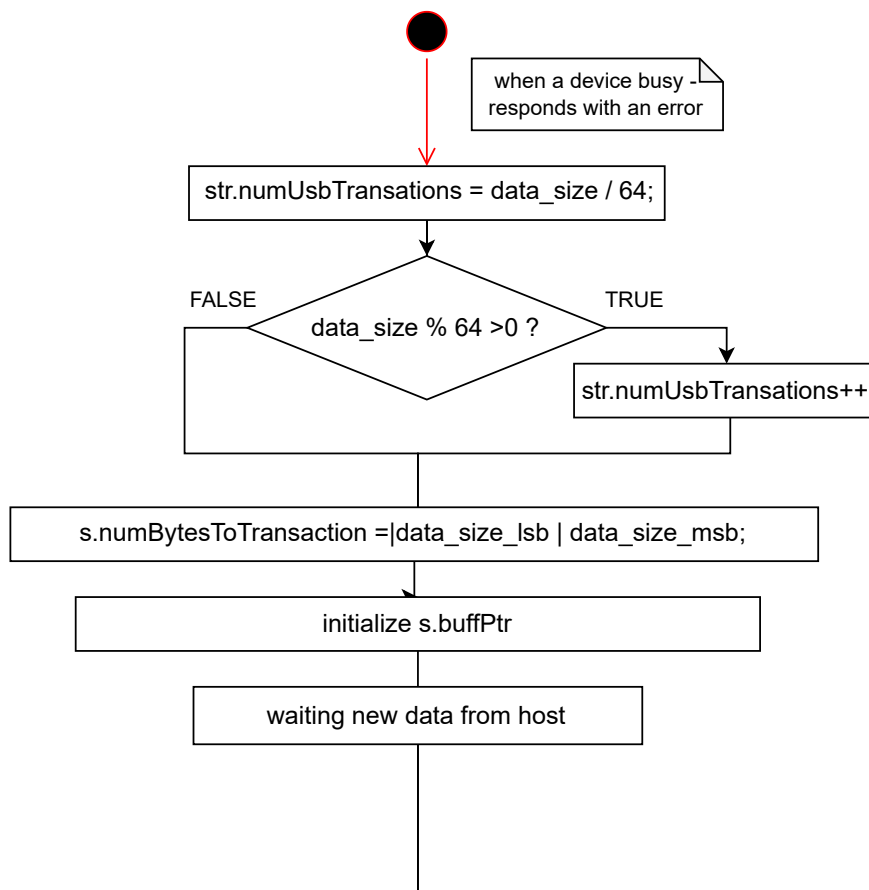
Commands	
a)write_to_i2c_dev	=24;
b)read_from_i2c_dev	=25;
c)reset_interface_i2c	=26;
d)setup_interface_i2c	=27;
e)read_last_stub_rx_i2c	=28;
f)write_tx_stub_buffer_i2c	=29;

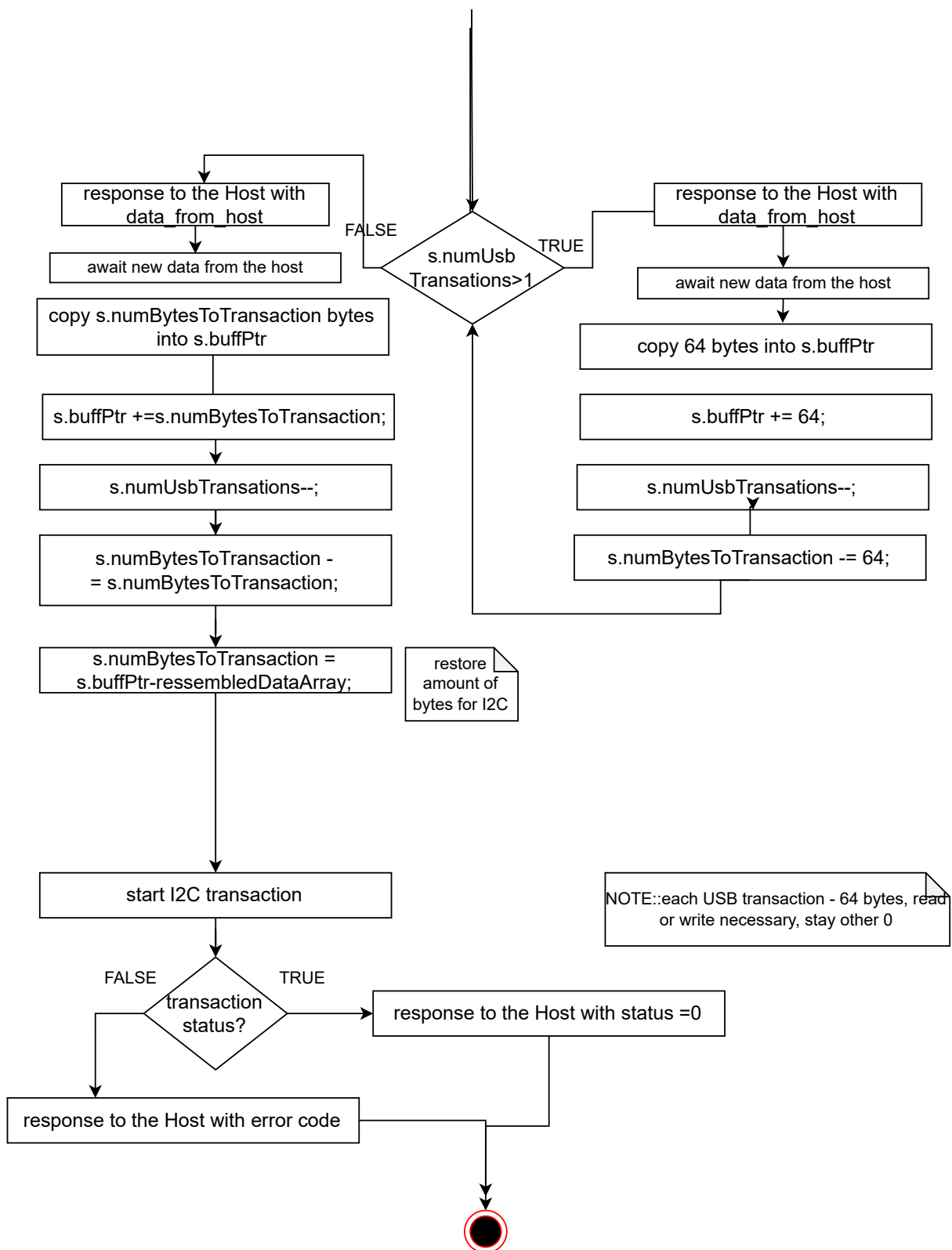
USB Device : the state machine

the structure must be aligned to 4 to improve CPU (Corex-M3) and peripheral performance

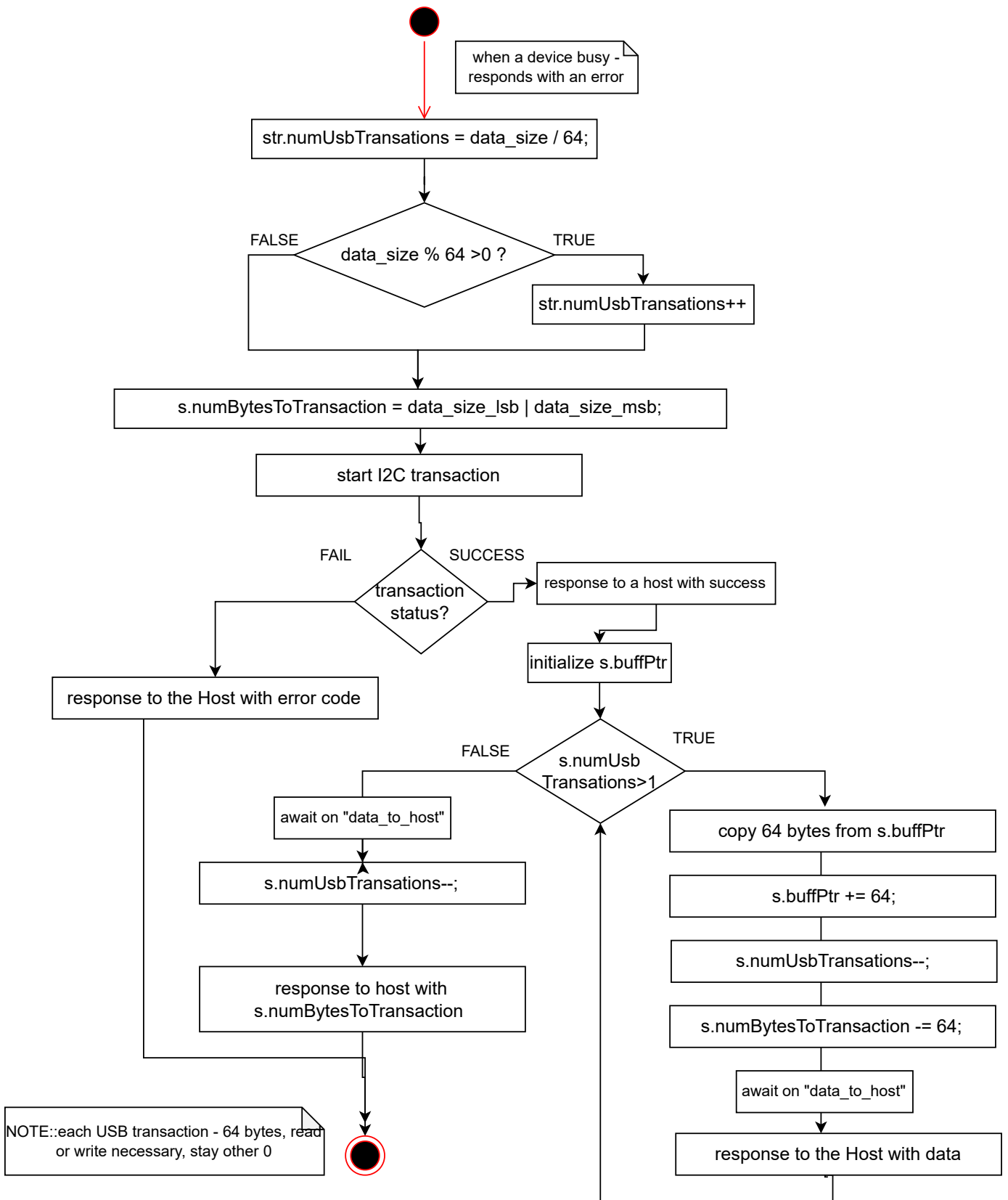
```
STRUCT {
    uchar typeOfAction;
    uchar slaveAddr;
    uchar* buffPtr; //changes during exec.
    uint16_t numUsbTransations;
    uint16_t numBytesToTransaction;
    uchar* reassembledDataArray; //not
    changed
}statesHandle;
```

CASE1: write_to_i2c_dev (device side)





CASE2: read_from_i2c_dev (device side)

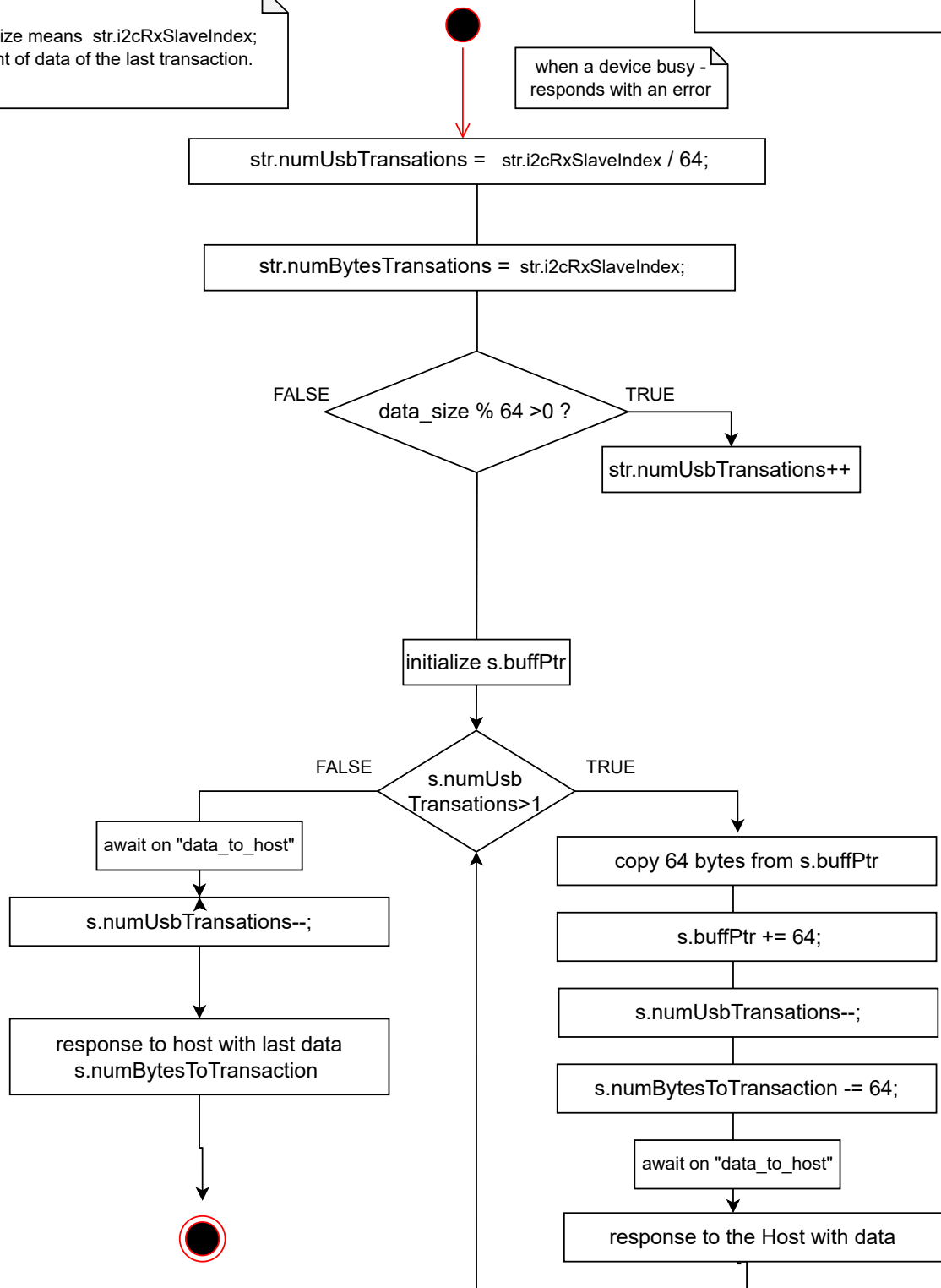


CASE3: read_last_internal_received_data (device side)

data_size means str.i2cRxSlaveIndex;
amount of data of the last transaction.

NOTE::there is I2C1 (slave) , that acts as
slave device with own address. This command
reads internal buffer - last received data.

when a device busy -
responds with an error

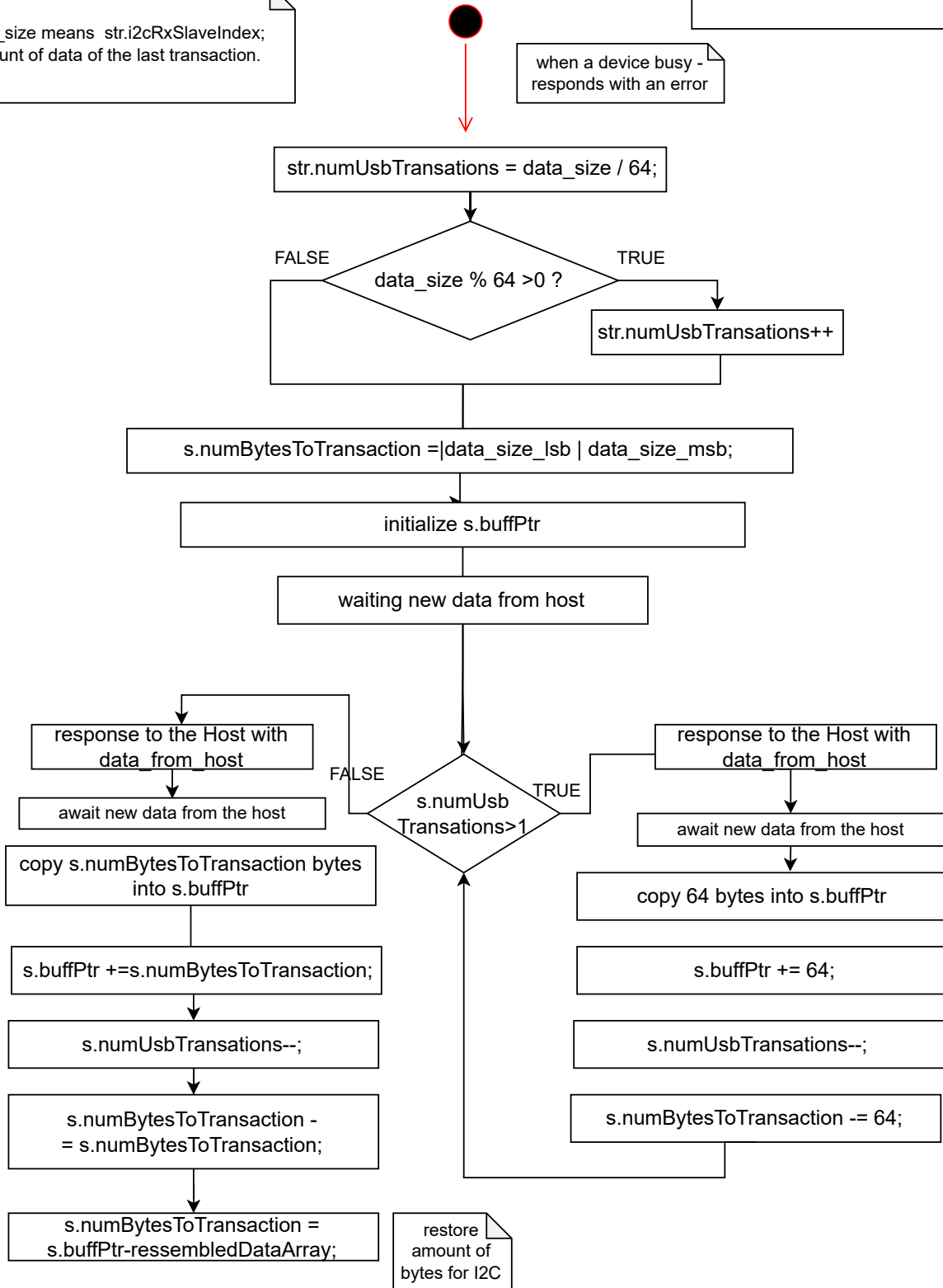


CASE4: Write slave Tx internal buffer(device side)

NOTE::there is I2C1 (slave) , that acts as slave device with own address. This command reads internal buffer - last received data.

data_size means str.i2cRxSlaveIndex;
amount of data of the last transaction.

when a device busy -
responds with an error



USB Device : the state machine

```

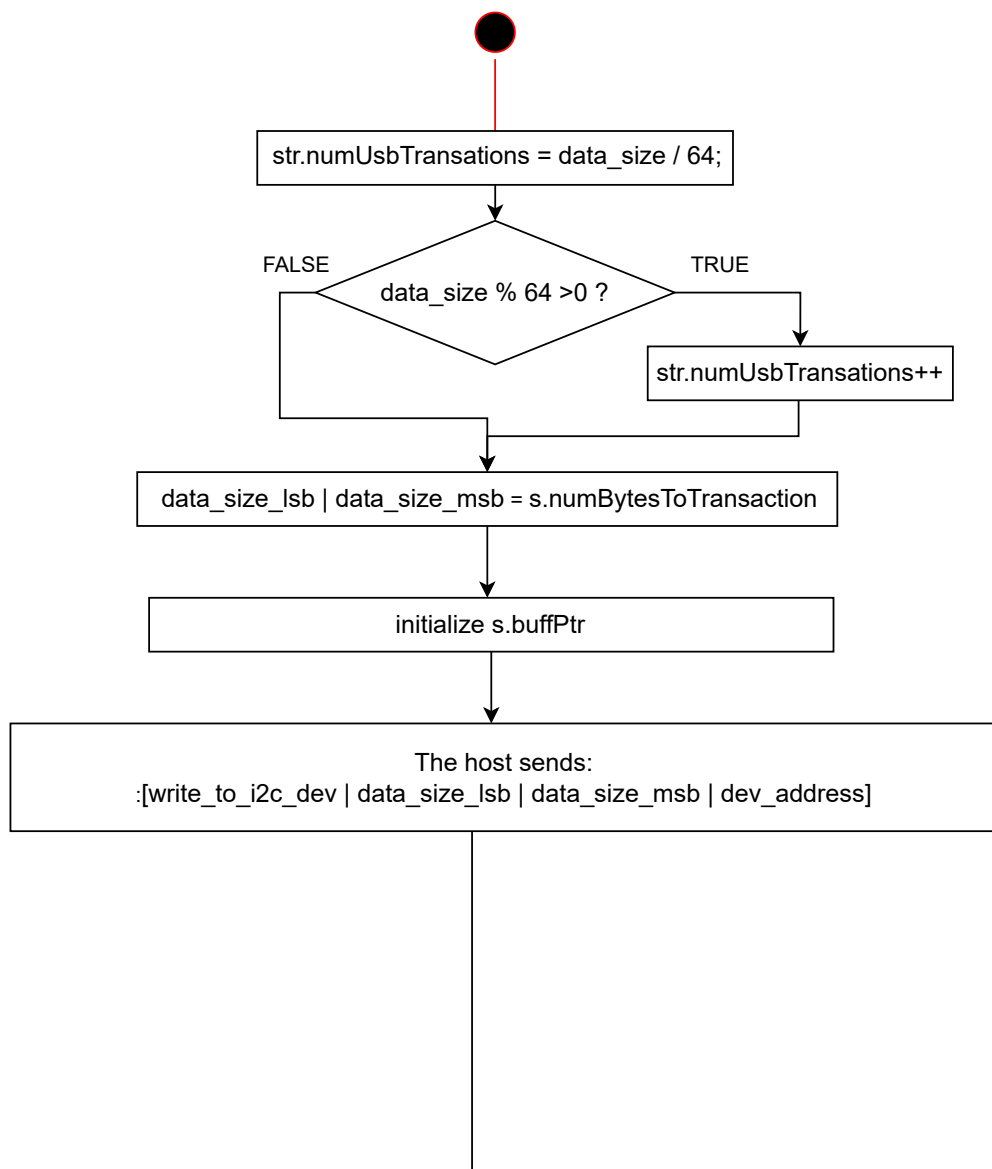
STRUCT {
  uchar typeOfAction;
  uchar slaveAddr;
  uchar* buffPtr; //changes during exec.
  uint16_t numUsbTransations;
  uint16_t numBytesToTransaction;
  uchar* reassembledDataArray; //not
  changed
}statesHandle;
    
```

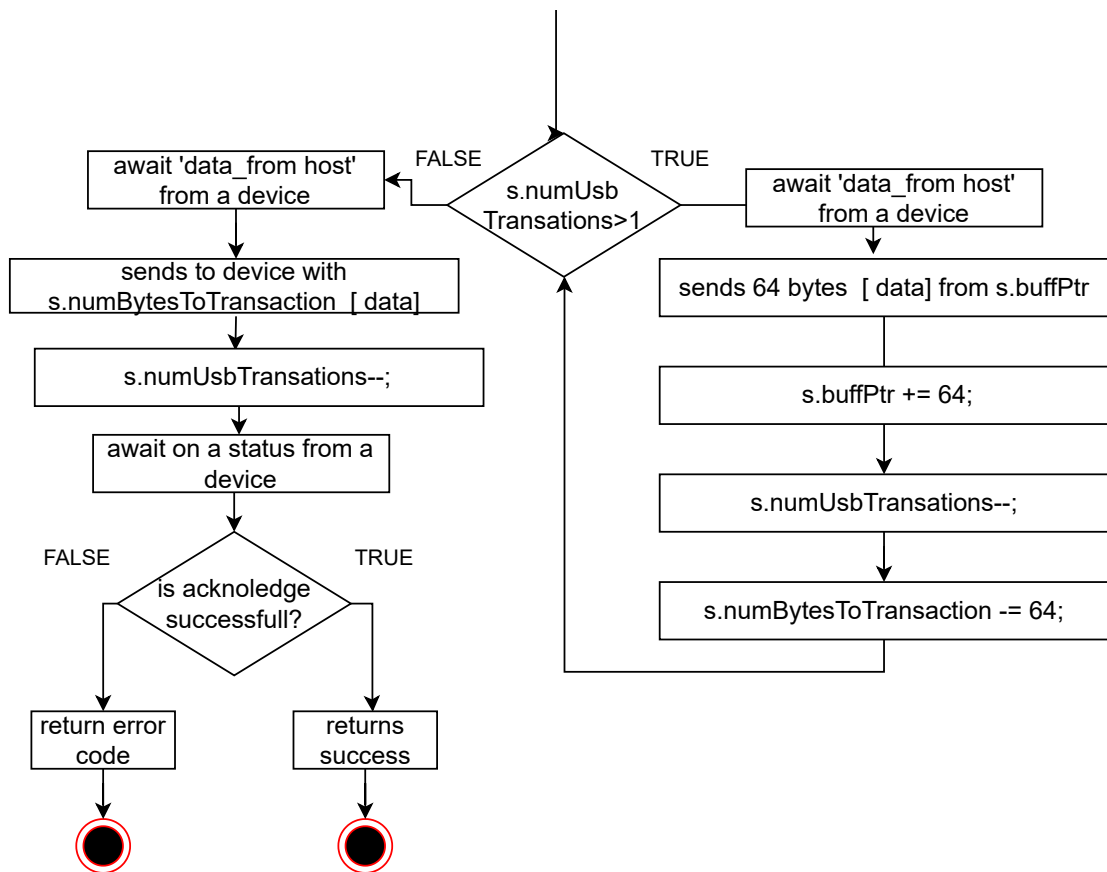
Error codes
a)Success =0;
b)Acknowledge failure AF =1;
c)Bus Error BERR =2;
d)Arbitration lost (ARLO) =3;
e)Overrun/underrun (OVR) =4;
f)Timeout =5;
g)Other error =6;
e)Busy =7;

Data markers
a)data_from_host =18;
b)data_to_host =19;

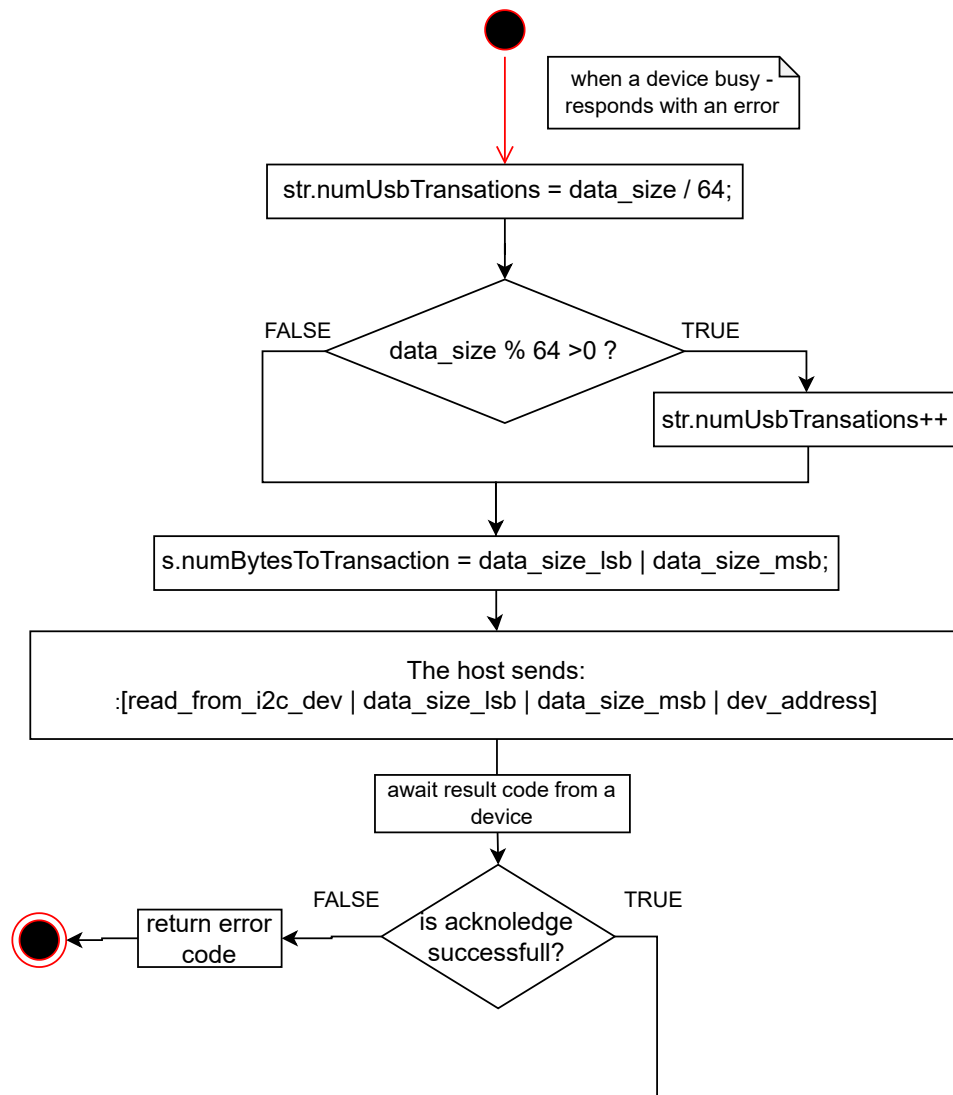
Commands
a)write_to_i2c_dev =24;
b)read_from_i2c_dev =25;
c)reset_interface =26;
d)setup_interface_i2c =27;
e)read_last_stub_rx_i2c =28
f)write_tx_stub_buffer_i2c = 29;

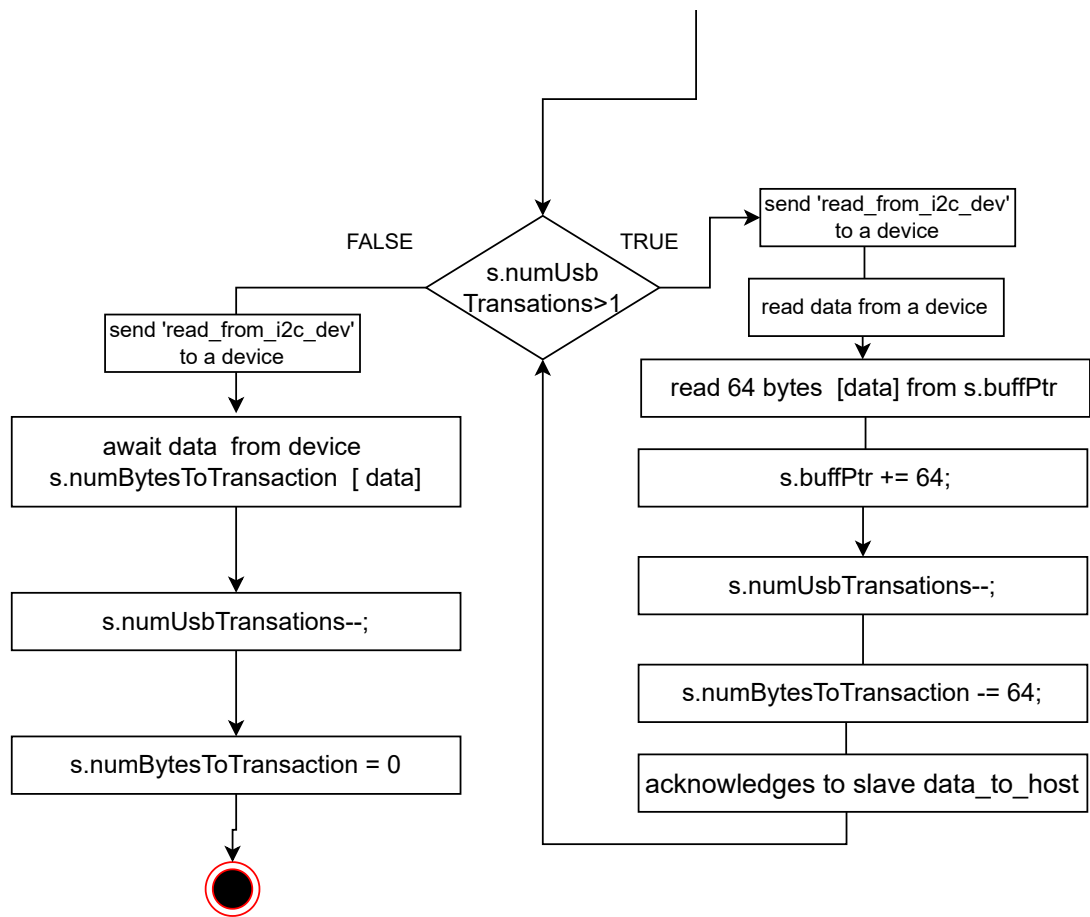
CASE1: write_to_i2c_dev (HOST side)



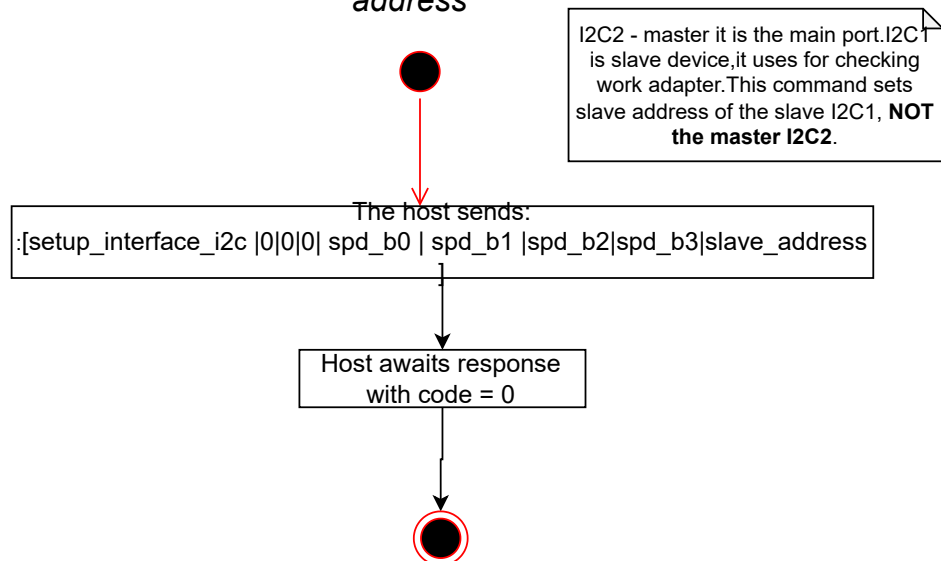


CASE2: read_from_i2c_dev (HOST side)



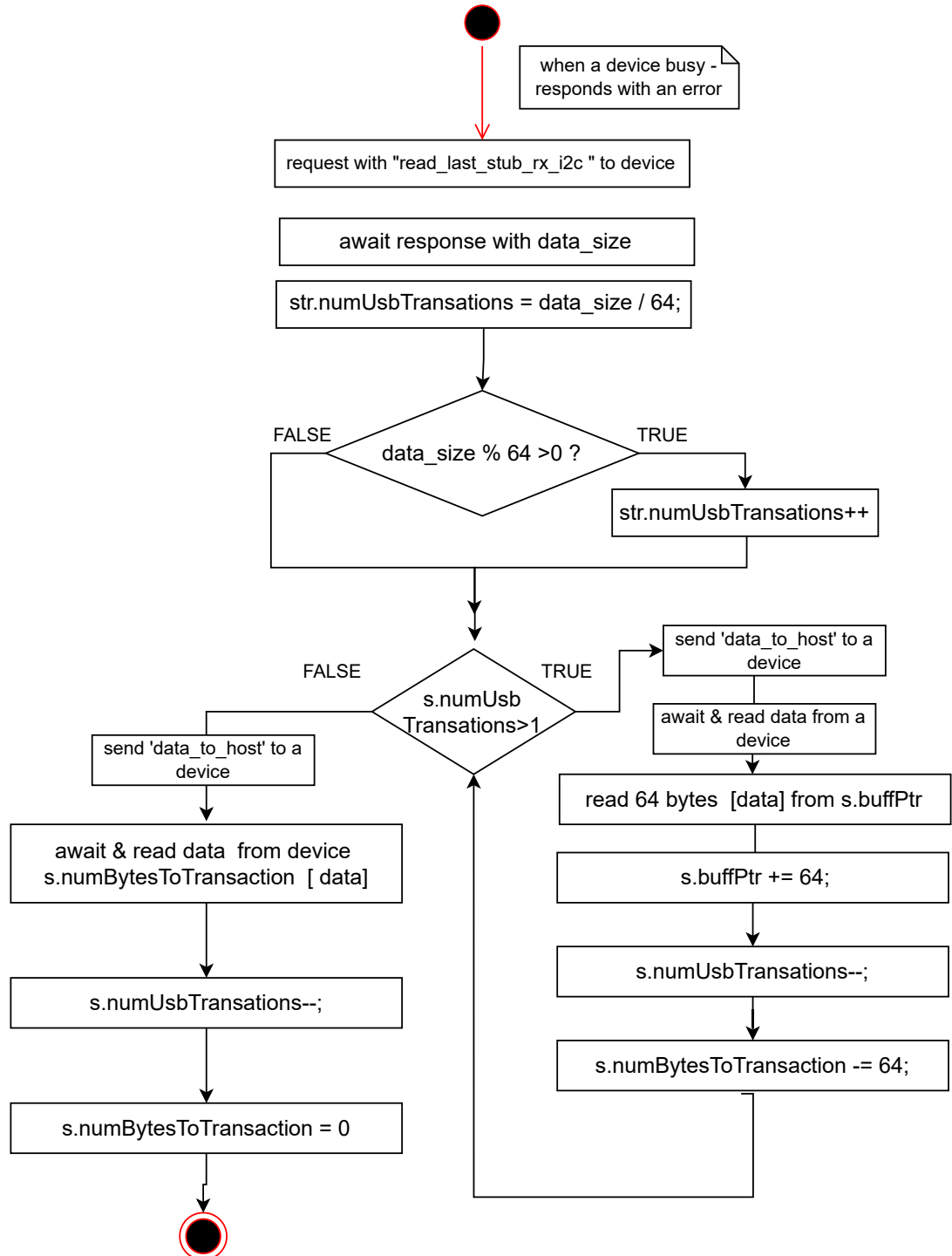


Set frequency and I2C1 slave address



CASE3: read_last_received_internal_data (HOST side)

I2C1 acts as slave device with it's own address. So, this comand reads last received data packet.



NOTE:
maximum size
of Rx/Tx slave
buffer =
256bytes. So,
avoid overflow

CASE4: write internal slave Tx bufer (I2C1) (HOST side)

