

FCM_kII DESIGN SCRATCH

COMMUNICATIONS

STANDARDIZED PORT NUMBERS
S. Listener (Broadcast): 65000 S. MISO: 65001 S. MOSI: 65002

BREAKDOWN	
Master	Slave
Parts	
<ul style="list-style-type: none">• Broadcast Thread<ul style="list-style-type: none">– Standard broadcast– Update broadcast– Shutdown broadcast• Listener thread NOTE: Here, responding Slaves can categorized are known and unknown<ul style="list-style-type: none">– List new, unknown Slaves for adding– Mark known, disconnected Slaves for reconnection– Ignore messages from connected Slaves– Send startup messages to Bootloaders when appropriate• Slave threads Depending on target Slave status.. DISCONNECTED: Wait for Slave to be marked by Listener thread AVAILABLE: Attempt handshake to connect. Mark as disconnected upon failure CONNECTED: Listen for messages, count timeouts when applicable, fetch commands from user	<ul style="list-style-type: none">• Listener thread: Listen for broadcasts. Depending on broadcast type... Standard: If disconnected, send reply; if connected, reset Master timeout counter. *Master timeout: Ping Master before assuming disconnection *Network timeout: Ping self before rebooting Update broadcast: Shutdown Processor and reboot Shutdown broadcast: Shutdown Processor and reboot Launch application: (For Bootloader) Ignore when in MkII; launch MkII when in Bootloader• MISO thread: Send updates to Master when connected:<ul style="list-style-type: none">– Fetch updates from Processor, if any, or send empty message to maintain connection– Send ping requests when flagged by Listener– Remain idle when disconnected. NOTE: Empty processor Queue• MOSI thread:<ul style="list-style-type: none">– Listen for messages from Master (when connected) and add them to Processor buffer– Reset timeout counter whenever a Message is received
Connection and Disconnection	
<ul style="list-style-type: none">• Use broadcast thread to keep Slave connected• Use MISO-side of Slave thread to listen for periodic Slave-side updates to know when to assume disconnection• NOTE: Send Disconnect message to Slave when assuming disconnection• Send multiple MOSI messages (use index)	<ul style="list-style-type: none">• Use listener thread to know if Master is still connected (based on broadcast)• Ping Master when considering disconnection• Send Disconnect message to Master when assuming disconnection• Ping self to check network status before assuming network error and rebooting• Also reset Master timeout counter upon reception in MOSI thread• Shutdown Processor when assuming disconnection from Master

MESSAGE FORMATS	
MOSI	MISO
Broadcast-side	
<ul style="list-style-type: none"> Standard broadcast: <u>N</u> PASSCODE <u>M.L.PORT</u> Update broadcast: <u>U</u> PASSCODE <u>M.L.PORT</u> <u>FILE_NAME</u> <u>FILE.SIZE.BYTES</u> Shutdown broadcast: <u>R</u> PASSCODE Launch MkII: <u>L</u> PASSCODE 	<ul style="list-style-type: none"> Standard broadcast reply (MkII): <u>A</u> PASSCODE <u>S_MAC</u> <u>N</u> <u>S_MISO.P</u> <u>S_MOSI.P</u> <u>VERSION</u> Error (MkII Listener): <u>A</u> PASSCODE <u>S_MAC</u> <u>E</u> ERROR.MESSAGE Error (Bootloader): <u>B</u> PASSCODE <u>S_MAC</u> <u>E</u> ERROR.MESSAGE Standard broadcast reply (Bootloader): <u>B</u> PASSCODE <u>S_MAC</u> <u>N</u>
Communications and Control	
<ul style="list-style-type: none"> Set DC: <u>MOSI_INDEX</u> <u>S</u> <u>D</u>:DC:00000000000000000000 Here each character in the string of zeroes corresponds to a fan in the target Slave's array. A '1' means the fan is to be set to the specified DC, and a '0' means it is to be left unchanged. Chase RPM: <u>MOSI_INDEX</u> <u>S</u> <u>C</u>:RPM:00000000000000000000 See "Set DC" for the meaning of the string of zeroes. Handshake: <u>0</u> <u>H</u> COMMS.CONFIG ARRAY.CONFIG COMMS.CONFIG is a comma-separated list with the following: <ol style="list-style-type: none"> MISO port MOSI port Period (ms) Max. Master timeouts <p>ARRAY.CONFIG is a space-separated list with the following:</p> <ol style="list-style-type: none"> Fan mode Num. active fans PWM frequency (Hz) Counter counts Pulses per rotation Max. RPM Min. RPM Min. DC Chaser tolerance (%) Max. fan timeout PWM pinout Tach. pinout Disconnect: <u>MOSI_INDEX</u> <u>X</u> Reboot: <u>MOSI_INDEX</u> <u>Z</u> Reset index: <u>MOSI_INDEX</u> <u>I</u> Slave will reset its MOSI index to 0. Ping: <u>MOSI_INDEX</u> <u>P</u> PSU: <u>MOSI_INDEX</u> <u>S</u> <u>W</u>:0 Here the last character will be a 1 or 0 depending on the desired PSU state (1 for ON and 0 for OFF). The PSU will be turned on upon startup and off upon shutdown and reboot by default. 	<ul style="list-style-type: none"> Maintain connection: <u>MISO_INDEX</u> <u>M</u> Sent to Master when there are no updates from Processor, but a MISO message is due to maintain connection. Standard update: <u>MISO_INDEX</u> <u>T</u> <u>DATA_INDEX</u> <u>RPMS</u> <u>DUTY_CYCLES</u> Here DUTY_CYCLES and RPMS are comma-separated lists of the DC and RPM values of each fan in the array, in order. Negative values will be used for RPMS of fans being "Chased." Error (MkII MISO): <u>MISO_INDEX</u> <u>E</u> ERROR.MESSAGE Handshakehspace.5emconfirmation: <u>MISO_INDEX</u> <u>H</u> For Slave-side exception handling and documenting. Ping request: <u>MISO_INDEX</u> <u>P</u> MISO index reset: <u>MISO_INDEX</u> <u>I</u> Master will reset its MISO index to 0.

Legend	
N “NORMAL” i.e. Standard broadcast U “UPDATE” i.e. Update broadcast R “REBOOT” i.e. Reboot MCU L “LAUNCH” i.e. Launch MkII S “STANDARD” i.e. Standard command for Processor D “DUTY CYCLE” i.e. Set Duty Cycle C “CHASE” i.e. Chase RPM H “HANDSHAKE” i.e. Handshake to start connection X “DISCONNECT” i.e. Assume disconnection (Shutdown Processor) Z “REBOOT” i.e. Reboot MCU I “INDEX” i.e. Reset MISO Index W “POWER” i.e. Power PSU	A “APPLICATION” i.e. Message from MkII B “BOOTLOADER” i.e. Message from Bootloader M “MAINTAIN” i.e. Maintain connection T “STANDARD” i.e. Standard update message E “ERROR” i.e. Error message P “PING” i.e. Ping request I “INDEX” i.e. MISO index reset

To Do
Mon. 6/25/18 - Tue. 6/25/18
<ol style="list-style-type: none"> 1. <u>Fix Bootloader 404 and empty file bugs</u> 2. Add missing pinout, PSU pins and external LED pins 3. Add placeholder for runtime pinout configuration 4. <u>Implement new message standard</u> Among other things... <ul style="list-style-type: none"> - Receive S.Error and B.Error messages in both Slave threads and listener thread - Use extra warnings in the event of a Bootloader error 5. <u>Implement Slave self-pinging</u>
Wed. 6/25/18 - Fri 6/29/18
<ol style="list-style-type: none"> 1. Implement runtime pinout configuration 2. Implement “efficient” tachometer 3. Implement Master-side firmware uploads 4. “Fully” modularize Master 5. Implement “verifications” and shutdown button 6. Implement user configuration and “null” settings 7. Implement multiprocessing
Mon. 7/2/18
<ol style="list-style-type: none"> 1. Fix PWM resolution 2. Fix Chaser 3. Fix RPM spikes (if applicable) 4. Fix Processor thread-safety
Tue. 7/3/18 - Fri. 7/6/18

1. + Fix S-side data types
2. + Implement index resets (including `dataIndex`!)
3. + Strong processor checks
4. + PSU auto on/off setting
5. + Input and output sockets in Master
6. + Implement hotkeys
7. + Implement plotter
8. + Document
9. + Add "help" section
10. + Compile Master-side
11. + Credits and licensing (And comments!)