FCMkII DESIGN SCRATCH

Communications

Standardized Port Numbers

S. Listener (Broadcast): `65000`

S. MISO: `65001`

S. MOSI: `65002`

Breakdown

Master

Parts

- Broadcast Thread
  - Standard broadcast
  - Update broadcast
  - Shutdown broadcast
- Listener thread
  NOTE: Here, responding Slaves can categorized are known and unknown
  - 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

Connection and Disconnection

- 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)

Slave

Parts

- 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:
  - 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:
  - Listen for messages from Master (when connected) and add them to Processor buffer
  - Reset timeout counter whenever a Message is received

Connection and Disconnection

- 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	
• Standard broadcast: N|PASSCODE|M_L_PORT • Update broadcast: U|PASSCODE|M_L_PORT|FILE_NAME|FILE_SIZE_BYTES • Shutdown broadcast: R|PASSCODE • Launch MkII: L|PASSCODE	• Standard broadcast reply (MkII): A|PASSCODE|S_MAC|N|S_MISO_P|S_MOSI_P|VERSION • Error (MkII Listener): A|PASSCODE|S_MAC|E|ERROR_MESSAGE • Error (Bootloader): B|PASSCODE|S_MAC|E|ERROR_MESSAGE • Standard broadcast reply (Bootloader): B|PASSCODE|S_MAC|N
Communications and Control	
• Set DC: MOSI_INDEX|S|D: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: MOSI_INDEX|S|C:RPM:00000000000000000000 See "Set DC" for the meaning of the string of zeroes. • Handshake: 0|H|COMMS_CONFIG|ARRAY_CONFIG COMMS_CONFIG is a comma-separated list with the following: 1. MISO port 2. MOSI port 3. Period (ms) 4. Broadcast period (ms) 5. Max. Master timeouts ARRAY_CONFIG is a space-separated list with the following: 1. Fan mode 2. Num. active fans 3. PWM frequency (Hz) 4. Counter counts 5. Pulses per rotation 6. Max. RPM 7. Min. RPM 8. Min. DC 9. Chaser tolerance (%) 10. Max. fan timeout 11. PWM pinout 12. Tach. pinout • Disconnect: MOSI_INDEX|X • Reboot: MOSI_INDEX|Z • Reset index: MOSI_INDEX|I Slave will reset its MOSI index to 0. • Ping: MOSI_INDEX|P • PSU: MOSI_INDEX|S|W: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 rebootby default.	• Maintain connection: MISO_INDEX|M Sent to Master when there are no updates from Processor, but a MISO message is due to maintain connection. • Standard update: MISO_INDEX|T|DATA_INDEX|RPMS|DUTY_CYCLES 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): MISO_INDEX|E|ERROR_MESSAGE • Handshakehspace.5emconfirmation: MISO_INDEX|H For Slave-side exception handling and documenting. • Ping request: MISO_INDEX|P • MISO index reset: MISO_INDEX|I Master will reset its MISO index to 0.

Legend	
N "NORMAL" i.e. Standard broadcast	**A** "APPLICATION" i.e. Message from MkII
U "UPDATE" i.e. Update broadcast	**B** "BOOTLOADER" i.e. Message from Bootloader
R "REBOOT" i.e. Reboot MCU	**M** "MAINTAIN" i.e. Maintain connection
L "LAUNCH" i.e. Launch MkII	**T** "STANDARD" i.e. Standard update message
S "STANDARD" i.e. Standard command for Processor	**E** "ERROR" i.e. Error message
D "DUTY CYCLE" i.e. Set Duty Cycle	**P** "PING" i.e. Ping request
C "CHASE" i.e. Chase RPM	**I** "INDEX" i.e. MISO index reset
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	

To Do

Mon. 6/25/18 - Tue. 6/25/18

1. Fix Bootloader 404 and empty file bugs
2. Add missing pinout, PSU pins and external LED pins
3. Add placeholder for runtime pinout configuration
4. Implement new message standard
   Among other things...
   - 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. **Implement Slave self-pinging**

Wed. 6/25/18 - Fri 6/29/18

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

1. Fix PWM `read()` precision
2. Fix PWM resolution
3. Fix Chaser
4. **Fix RPM spikes (if applicable)**
5. 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!)