FarmShield Firmware Outline

The spears will run an nRF52840 microprocessor which will require the following hardware drivers to read and control the various peripherals.

Shield HAL Drivers

- SPI driver
- I2C Driver
- UART
- ADC
- EEPROM

Shield Peripheral Application Layer

All the peripherals on the shield MCU will require modules atop the hardware abstraction layer. These modules will be made modular so that the code may be reusable regardless of any hardware changes in the future.

The following are the modules required for the Shield.

- Battery Management stack
- Modem Stack
- Firmware OTA stack
- Configuration storage and retrieval stack
- SD card management stack
- Radio Stack
- Data over radio TX/RX stack
- Message Queue and encryption stack
- Relay Control Stack
- Motor Controller stack
- Water Flow Meter stack
- Valve Control Stack
- Light Level Sensor stack

Shield System Application Layer

This section covers how all the various stacks come together to follow a routine of how the device works according to a process flow. This includes receiving and bundling together data from the various spears paired to the shield, encrypting and transmitting data to the cloud. The system application layer will also handle how the device interacts with the cloud and users.

Messaging Protocol

Maximum length of data is 256 bytes over MQTT protocol

Start	length	Msg type	direction	s/n	msg id	payload	end
0x3C	1 byte	1 byte	0x55/0x44	12 bytes	1 byte	variable	0x3E

Msg type

value	Type
0x44	Sensor data msg
0x53	Settings msg
0x45	Error msg
0x41	Message ACK
0x00	Unknown type

direction

value	direction
0x55	Upload to Server
0x44	Download from server

if message is of type **Message ACK** , **payload** section will not be available

Example serial number (**s/n**) in hex
`e00fce689a754705e79a0e37` split into bytes like so
`0xe0,0x0f,0xce,0x68,0x9a,0x75,0x47,0x05,0xe7,0x9a,0x0e,0x37`

msg id increments and overflows to start over from zero to prevent multiple processing of the same message

payload

Message payload is packaged in form of a string of tuples in the followinf format

`<tuple_id><length><data_bytes><tuple_id><length><data_bytes>...`

tuple member	size
tuple id	1 byte
length	1 byte
data bytes	Specified by **length** above

Currently supported tuple ids and their length

Tuple Name	Tuple ID	Length	Description
MSG_ACK	0x00	1	

Tuple Name	Tuple ID	Length	Description
SPEAR_ID	0x01	12	
STORE_TIMESTAMP	0x02	4	
SEND_TIMESTAMP	0x03	4	
SOIL_MOISTURE	0x04	2	
AIR_HUMIDITY	0x05	2	
SOIL_HUMIDITY	0x06	2	
WATER_DISPENSED	0x07	4	floating point value of water dispensed in liters
CARBON_DIOXIDE	0x08	2	
AIR_TEMPERATURE	0x09	2	
SOIL_TEMPERATURE	0x0A	2	
SOIL_NPK	0x0B	2	
LIGHT_INTENSITY	0x0C	2	
SHIELD_BATTERY_LEVEL	0x0D	4	
SPEAR_BATTERY_LEVEL	0x0E	2	floating point value of % state of charge in shield
VALVE_POSITION	0x0F	1	
IGH_SEND_SETTINGS	0x10	Variable	
IGH_READ_SETTINGS	0x11	Variable	
SPEAR_DATA	0x12	Variable	
SPEAR_RF_ID	0x13	2	
SHIELD_RF_ID	0x14	2	
SEND_INTERVAL	0x15	4	
OP_STATE	0x16	1	
SHIELD_ID	0x17	12	
SPEAR_BATT_LOW_THRESHOLD	0x18	2	
SHIELD_BATT_LOW_THRESHOLD	0x19	2	
BUTTON_PRESS	0x19	1	How long button was pressed in seconds
RESTART	0xFD	1	
DATA_PKT	0xFE	Variable	

Tuple Name	Tuple ID	Length	Description
END_OF_PKT_ID	0xFF	-	

IGH_READ_SETTINGS

In order to get the current settings in a device, the cloud platform must send a message with and **IGH_READ_SETTINGS** tuple in the payload.
The read settings tuple should be in the following format:
`<IGH_READ_SETTINGS><length><Settings_subid><Settings_subid>...`
The settings subid are listed in the table below.

The device will respond with the requested settings in the next payload in the following format:
`<IGH_READ_SETTINGS><total_length><Settings_subid><length><data><Settings_subid><length><data>...`

SETTINGS SUBID TUPLES

Subid Name	Tuple ID	Length
SUBID_OPSTATE	0x01	1
SUBID_REPORTING_INTERVAL	0x02	4
SUBID_DATA_RESOLUTION	0x03	4
SUBID_SET_SERIAL_NUMBER	0x04	12
SUBID_MQTT_BROKER	0x05	Variable
SUBID_MQTT_BROKER_PORT	0x06	2
SUBID_TIMEZONE	0x07	1
SUBID_IRRIGATION_HR	0x08	1
SUBID_WATER_DISP_PERIOD	0x09	4
SUBID_MQTT_USERNAME	0x0A	Variable
SUBID_MQTT_PASSWORD	0x0B	Variable
SUBID_WATER_AMOUNT_BY_BUTTON	0x0C	4
SUBID_SOIL_MOISTURE_LOW	0x10	2
SUBID_AIR_HUMIDITY_LOW	0x11	2
SUBID_SOIL_HUMIDITY_LOW	0x12	2
SUBID_CARBON_DIOXIDE_LOW	0x13	2
SUBID_AIR_TEMPERATURE_LOW	0x14	2
SUBID_SOIL_TEMPERATURE_LOW	0x15	2

Subid Name	Tuple ID	Length
SUBID_SOIL_NPK_LOW	0x16	2
SUBID_LIGHT_INTENSITY_LOW	0x17	2
SUBID_SHIELD_BATTERY_LEVEL_LOW	0x18	2
SUBID_SPEAR_BATTERY_LEVEL_LOW	0x19	2
SUBID_DAILY_WATER_DISPENSED_MIN	0x1A	4
SUBID_SOIL_MOISTURE_HIGH	0x30	2
SUBID_AIR_HUMIDITY_HIGH	0x31	2
SUBID_SOIL_HUMIDITY_HIGH	0x32	2
SUBID_CARBON_DIOXIDE_HIGH	0x33	2
SUBID_AIR_TEMPERATURE_HIGH	0x34	2
SUBID_SOIL_TEMPERATURE_HIGH	0x35	2
SUBID_SOIL_NPK_HIGH	0x36	2
SUBID_LIGHT_INTENSITY_HIGH	0x37	2
SUBID_SHIELD_BATTERY_LEVEL_HIGH	0x38	2
SUBID_SPEAR_BATTERY_LEVEL_HIGH	0x39	2
SUBID_DAILY_WATER_DISPENSED_MAX	0x3A	4

IGH_SETTINGS

The **IGH_SETTINGS** tuple is used to send new settings down to the device in the following byte stream format using the SUBIDs listed above.

```
<IGH_SEND_SETTINGS><total_length><Settings_subid><length><data><Settings_subid>
<length><data>...
```

MQTT Protocol

Each IGH device shall connect to the MQTT broker `farmshield.illuminumgreenhouses.com` at port `1883`. The following are credentials for this broker:

```
Username = shields
Password = 940610b43b1
```

The devices will use their serial number in hex string as their IDs and subscribe to a download topic but publish to a data topic.
The topics shall be the device serial number with either `44` for download or `55` for upload appended at the beginning of the string.

For example;

Downlaod Topic
`44e00fce689a754705e79a0e37`

Upload Topic
`55e00fce689a754705e79a0e37`

for device with serial number `e00fce689a754705e79a0e37`.

`(All the parameters can be updated via settings except for the Download and Upload topics which are automatically generated by the device.)`

All messages sent to the device must be channeled throu the respective download topic and the device must send any payload through its respective
Upload topic.
The Messages published over MQTT must be in the format expressed above.

ERRORS/EVENTS TO CONSIDER

1. Not/connected to cloud
2. Not/connected to Network
3. Not/connected to MQTT broker
4. Battery Low
5. SD Card Fault
6. Unreachable Spear
7. Button Press

WATER IRRIGATION LOGIC

Button Control Logic

- Holding the button down for at least ***FIVE SECONDS*** will open the valve. If *_**100 litres of water**_ pass through the meter **OR** if *_**30 minutes**_ elapse since the valve opened, the valve will automatically close. Opening the valve via the button automatically disables anu form of automatica irrigation
- During Normal operation, the system will wait for *_**0600Hrs**_ before irrigation can start. At this hour, the system will first dispense *_**300 litres of water**_ after which the system moves into **SENSOR DISPENSE MODE**. The soil humidity data is then used to open and close the vlave. If the humidity readings are between *_**60%**_ and *_**90%**_, the valve will open and if *_**100 litres of water**_ pass through the meter **OR** if *_**30 minutes**_ elapse since the valve opened, the valve will automatically close.
- Holding the button for ***TWO SECONDS*** will disable **SENSOR DISPENSE MODE** and hodling the button down for ***TWO SECONDS*** once more re enables **SENSOR DISPENSE MODE**.

- If the total water dispensed in a single day is above *<strong><em>800 litres of water</em></strong>, **SENSOR DISPENSE MODE** is disabled but water can still be dispensed via a ***FIVE SECONDS*** button press.

NOTES:
*****100 litres of water*** *- Default value, can be changed using settings* ***SUBID_WATER_AMOUNT_BY_BUTTON***
*****30 minutes*** *- Default value, can be changed using settings* ***SUBID_WATER_DISP_PERIOD***
*****300 litres of water*** *- Default value, can be changed using settings* ***SUBID_DAILY_WATER_DISPENSED_MIN***
*****800 litres of water*** *- Default value, can be changed using settings* ***SUBID_DAILY_WATER_DISPENSED_MAX***
*****0600Hrs*** *- Default value, can be changed using settings* ***SUBID_IRRIGATION_HR***