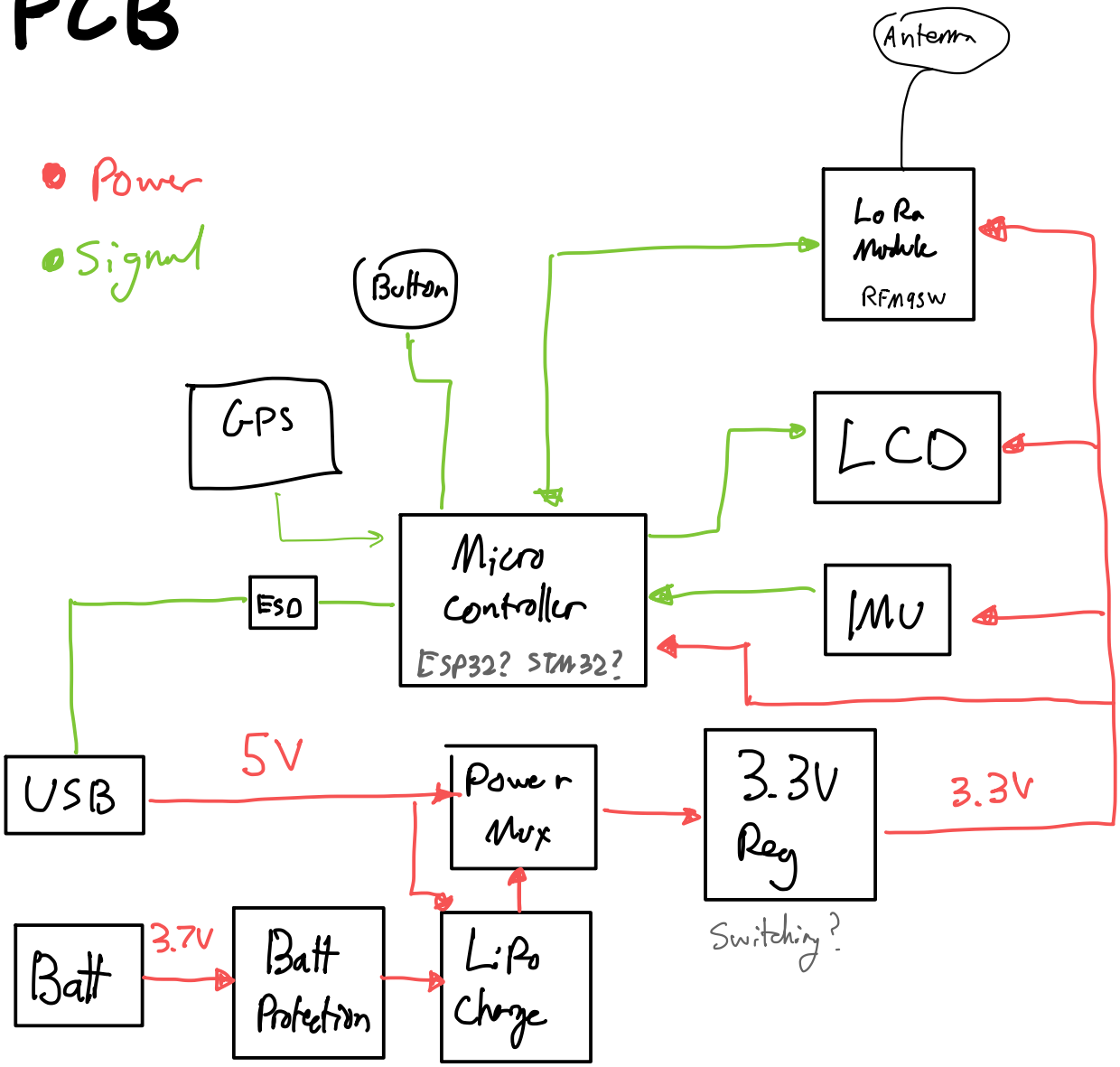
PCB

- Power
- Signal



- Need LCD driver board?
- More UI devices?
- Additional supporting circuitry for K?

Power Draw

Battery: 1s LiPo → 3.7 V ; 500 mAh

Target battery life: 3-5 hrs

↳ Acceptable Avg. Current Draw: 100 - 160 mA

Regulator Efficiency:

Microcontroller:

IMU:

Screen:

GPS:

Lora Module:

RX: 12 mA

TX:
- +20 dBm: 120 mA
- +17 dBm: 87 mA
- +13 dBm: 29 mA
- +7 dBm: 20 mA

Sleep: 0.2 mA

Idle/Standby: 1.5 mA

w/o idle mode

Let n = # paired devices

$$I_{avg} = \frac{1}{n} I_{Tx} + \frac{n-1}{n} I_{RX}$$

For 3 devices: n = 3

$$I_{avg} = \frac{1}{3}(120\ mA) + \frac{2}{3}(12\ mA)$$

$$= \boxed{48\ mA\ @\ +20\ dBm}$$

w/ idle mode

Worst Case Time on Air: 400 ms

Max n: 10

$I_{avg,active} = 48\ mA$

$T_{active} = n \times ToA = 10 \times 400\ ms = 4\ s$

$T_{idle} = Update\ Period - T_{active}$

$\approx 60\ s - 4\ s = 56\ s$

$$I_{avg} = I_{active} \times \frac{T_{active}}{Update\ Period} + I_{idle} \times \frac{T_{idle}}{Update\ Period}$$

$$= 48\ mA \times \frac{4}{60} + 1.5\ mA \times \frac{56}{60}$$

$$\boxed{= 4.6\ mA}$$

MCU Selection

	STM32 F401RCT6TR	STM32 L476RETG	ESP32 S3-WROOM-1-N8	ESP32 C3-WROOM-02-H4	RP2040
$	$5.85	$6.34	$3.2	$2.00	$1.00
RAM	64k	128k	512k	400k	264k
Flash	256k	512k	8M	4M	...
Freq	84 Mhz	80 Mhz	240 MHz	160 MHz	133 MHz
Light sleep	2.4 µA	120 nA	8 µA	130 µA	...
Typ	128 µA/MHz	100 µA/MHz	≈ 208 µA/MHz	175 µA/MHz	280 µA/MHz
IO Count	81	114	45 (Programmable)	16	30
SPI	4	3	2	3	
I²C	3	3	2	2	
W:F:/BT			X	X	

* Req. Ext. Flash

GPIO / Port Map

IMU:

VCC
GND
SCL / SCLK : O
SDA / SDI
~~EDA~~
~~ECL~~
ADO / SDO
INT : Interrupt DIO
NCS : Chipsel
~~FSYNC:~~

Screen:

RST
CS
DO
SDA
SCL : O
GND
VCC

Lora:

VIN
GND
EN
GO: Interrupt
Sck : O
MISO
MOSI
CS
RST

GPS

VCC
GND
TX
Rx
} UART

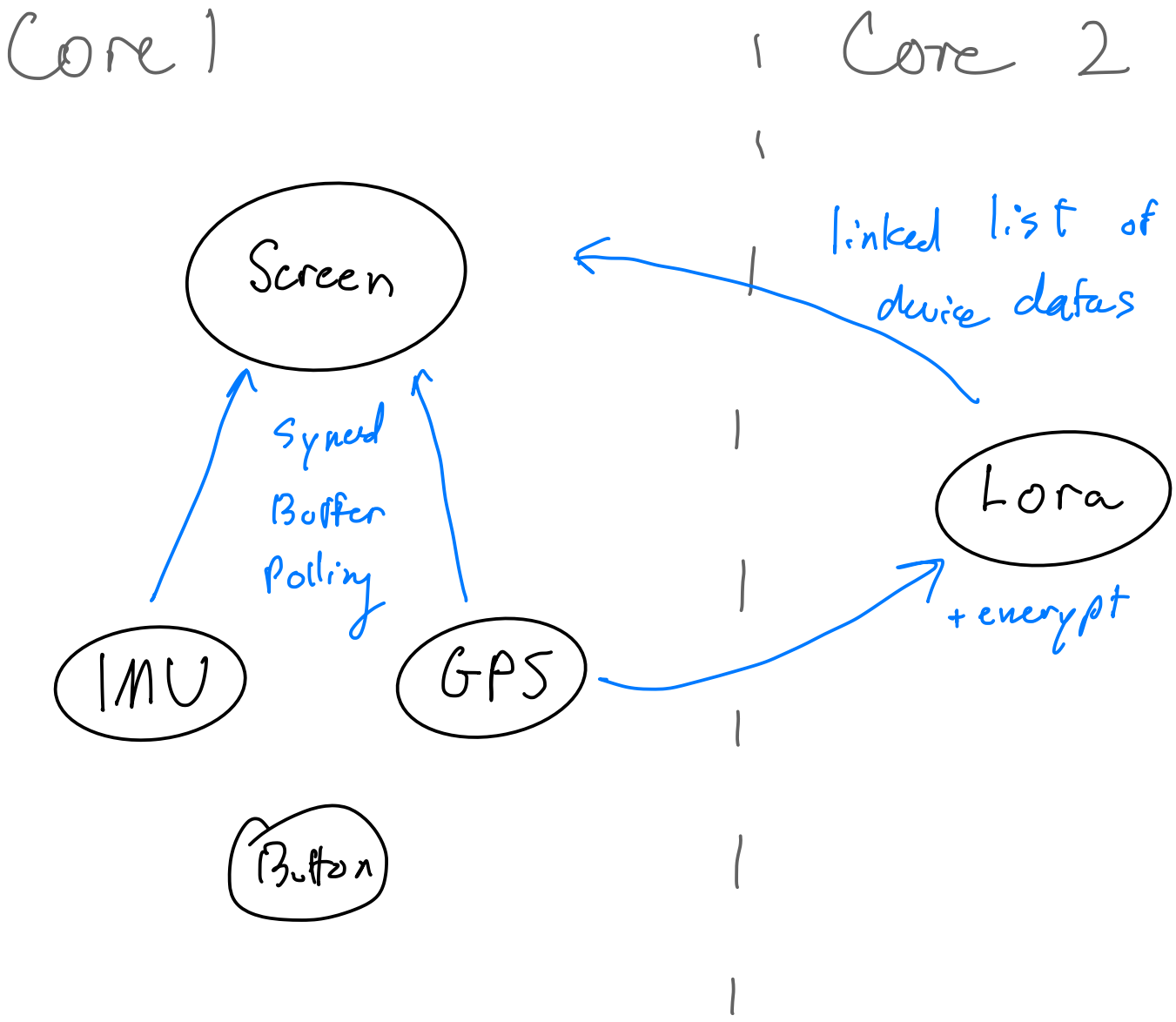
UI:

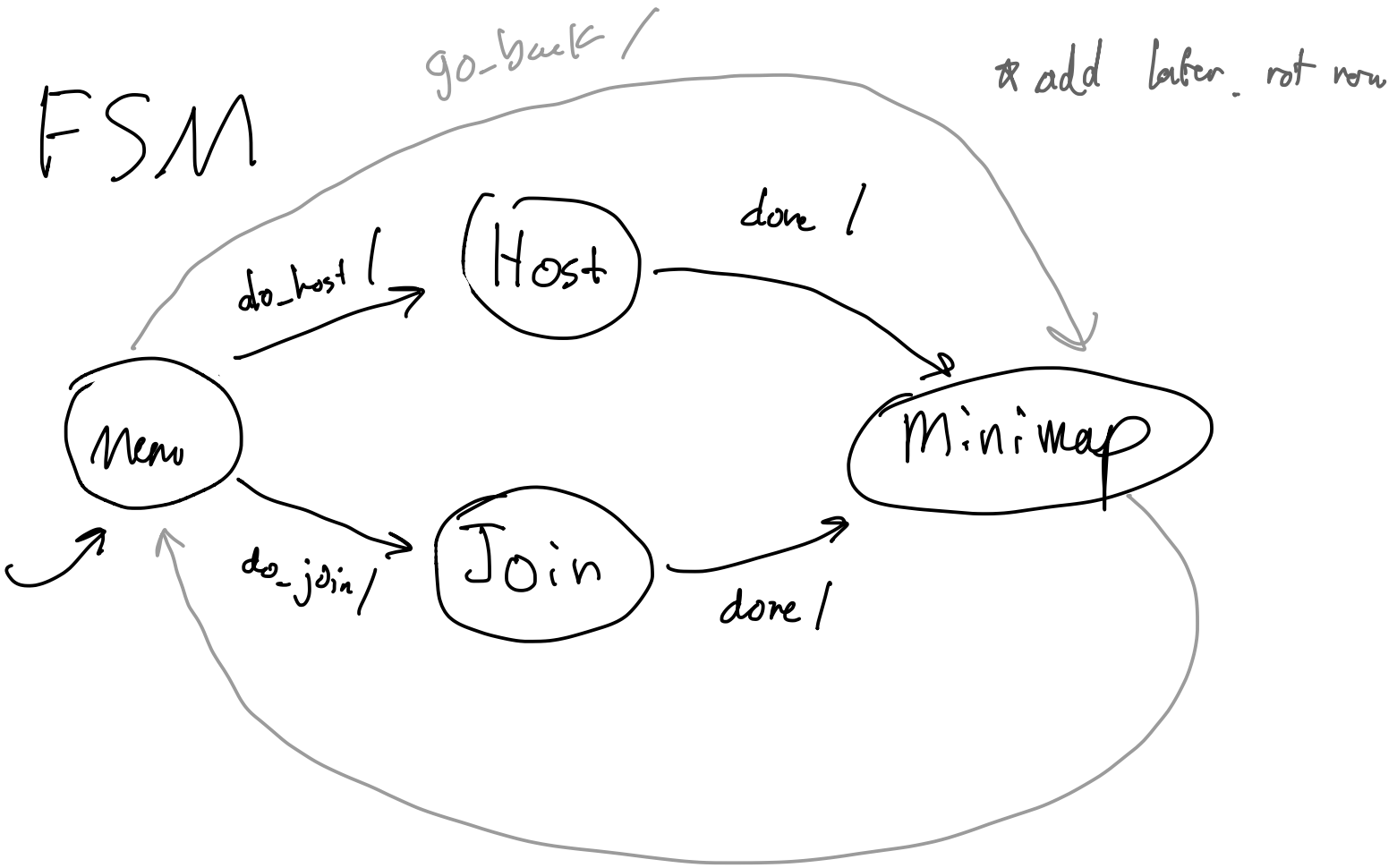
Button x2

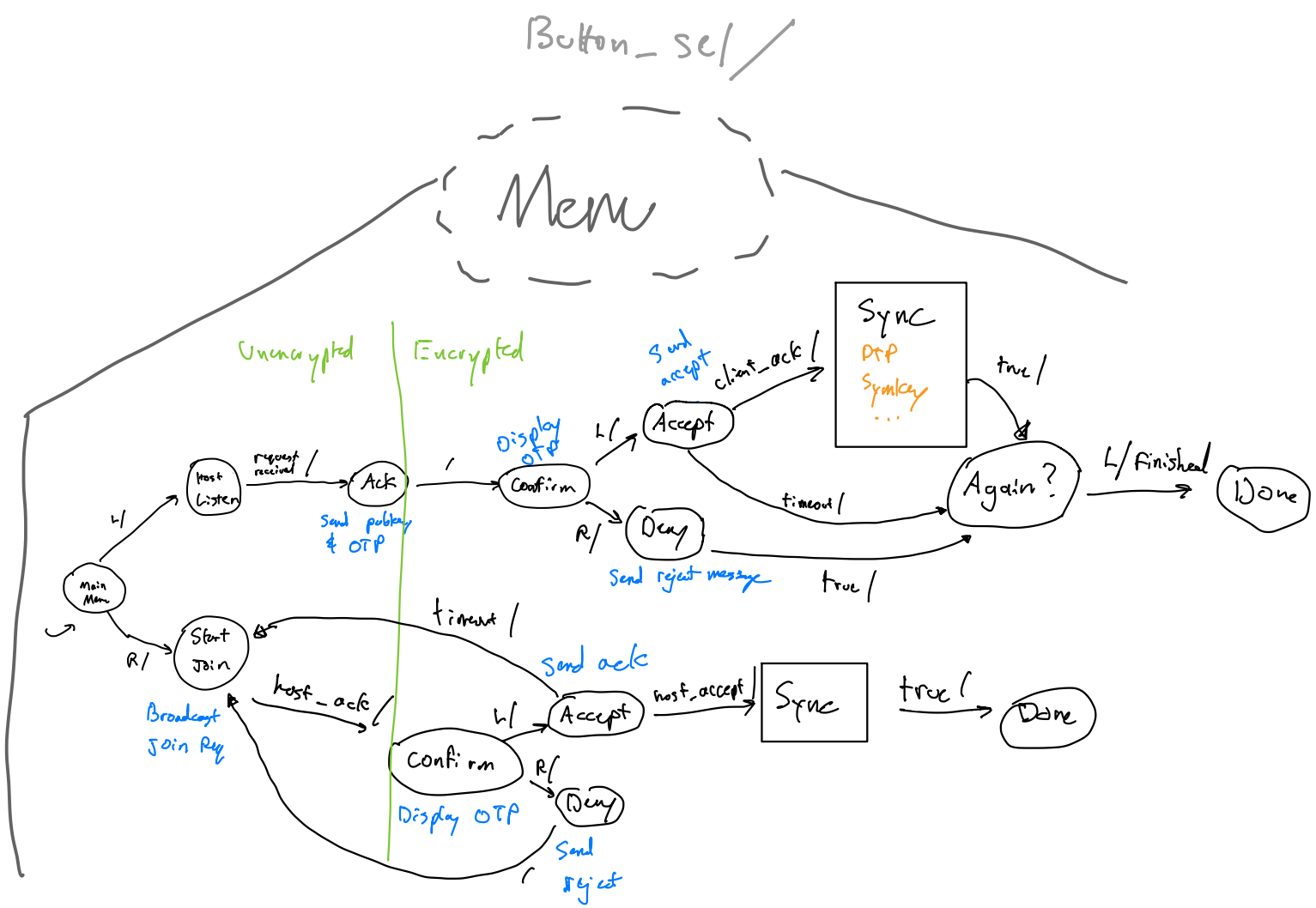
Figure Out

- Tasks
- Communication
- FSM
- Interrupts?

Data Flow

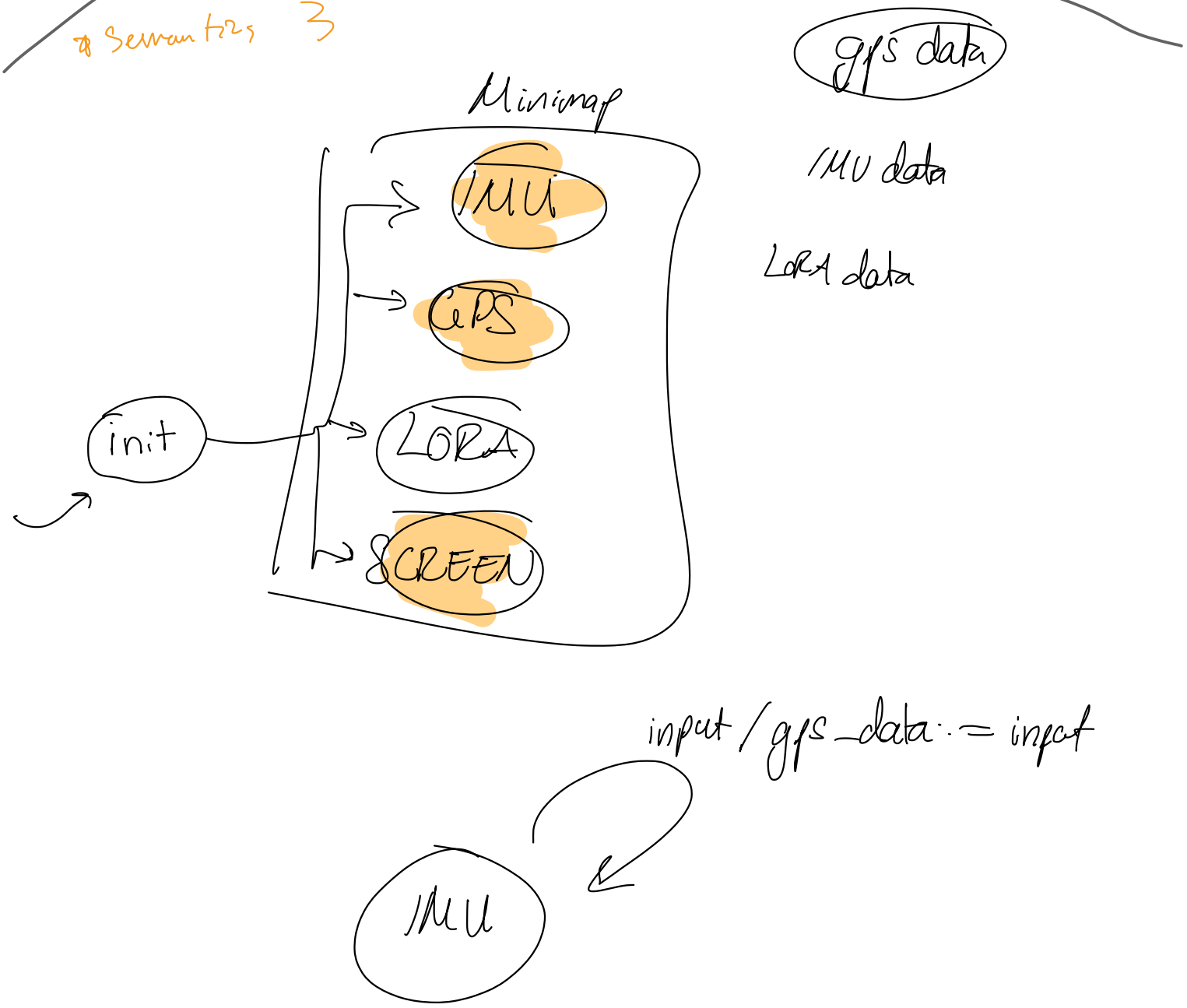



FSM
go_back /
* add later_ not now
Menu
do_host /
Host
done /
Minimap
do_join /
Join
done /


Button_sel /
Menu
Unencrypted
Encrypted
Main Menu
L/
Host Listen
request received /
Ack
Send publkey & OTP
/
Confirm
Display OTP
L/
Accept
Send accept
client_ack /
Sync
DH
Symkey
...
true /
Again?
L/ Finished
Done
timeout /
R/
Deny
Send reject message
true /
Start Join
R/
Broadcast Join Req
timeout /
Send ack
host_ack /
L/
Accept
host_accept
Sync
true /
Done
Confirm
R/
Deny
Display OTP
Send Reject

Minimap

* Servants 3



input / gps_data := input

Screen

global state { hosting, joining, normal }

Core task

```
while 1:
    switch (state):
        host_stuff
        join_stuff
```