Smoke Signal
MES Final Project

Bee Goethals

Smoke Signal

Application description

Smoke Signal is a unit that tracks concentration of gas in relation to a point of origin. It rolls around avoiding obstacles and following walls, recording its position.

Application Theory

1. On turn on SS determines current point of origin
2. Using gas detection sensors SS will measure the concentration at each location
3. The wheels and wall following algorithm allows SS to be autonomous in movement and detection on a 2D plane
4. Transmission/uploading to the back end occurs after measuring, with the location, origin and gas concentration data

Application MVP

Mapping unit for determining concentration of a gas in a given area. For the first iteration the mapping will be a simple plane calculation from the point of origin, with all data transmitted over BLE. Needs for a buffer will be determined during testing.

- PWM control for motors
- Gyro/Accel for movement
- SO4 sensor for current ppm concentration
- BLE for communication

Hardware Description

Smoke Signal has 2 motors, a piecemeal of batteries and development boards to create this MVP. SS isn't trying to win beauty contests, SS wants to discover the world according to sensors. This empiricism driven bot is programmed through ST-LINK, but longs to use JTAG.

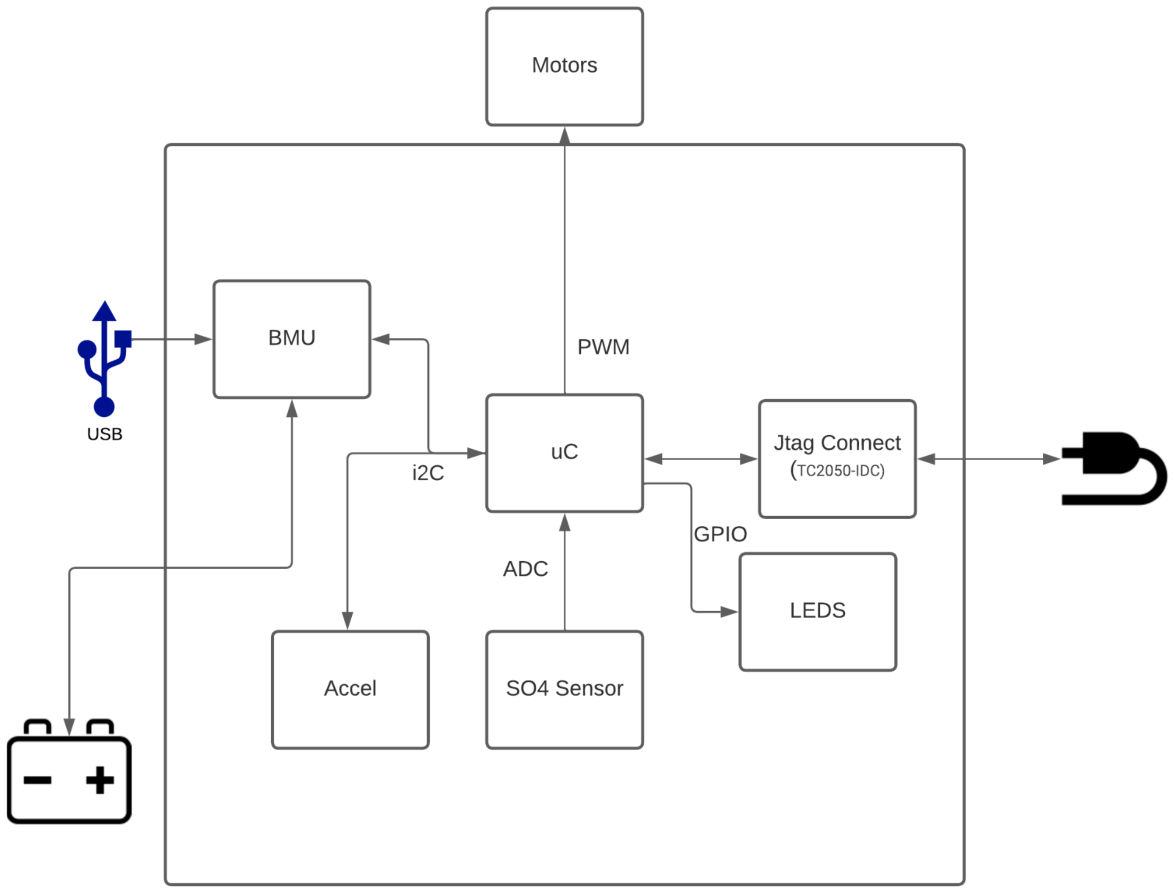

Diagram 1: Hardware Block Diagram

Software Description

Code in general - Git link TBD

The code is a state machine that runs on the STM32F411 development board.

Parts that were written (3-5 sentence per module)

TBD

Code reused and licenses

- CLI from reusable

- STM32CubeIDE - auto generate uart/CLI/interrupts
-

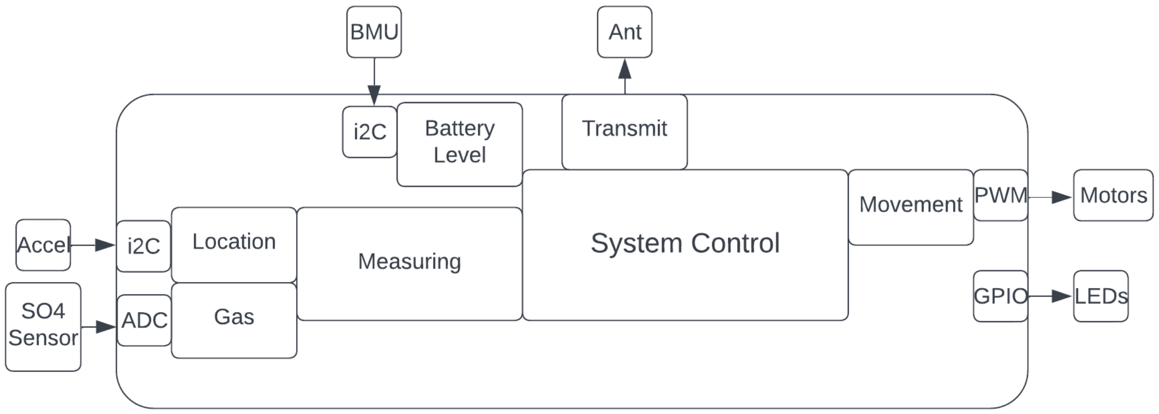

Diagram 2: Software Block Diagram

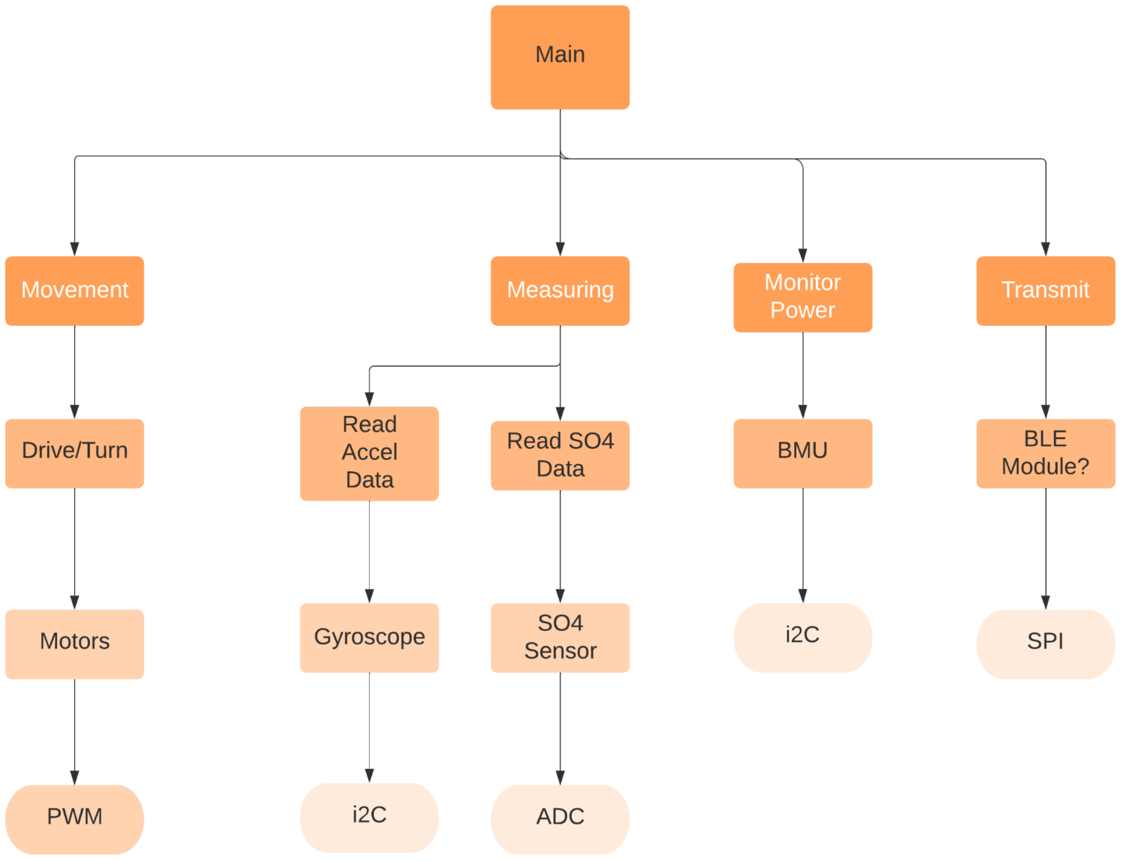

Diagram 3: Hierarchy of Control

State	**Special action**	**LED**	**Timeout**	**Bat < 20**	**Completed Map**	**Repeated Point**	**Right Path Open**	**Blocked Path**
INIT	POST - check	Blink Green	POINT OF ORIGIN	x	x	x	x	x
POINT OF ORIGIN	Save Origin Point	Blue	LOCATION DETECTION	SHUTDOWN	x	x	x	x
LOCATION DETECT	x	Blink Blue	FORWARD	SHUTDOWN	SHUTDOWN	MOVE IN	RIGHT TURN	LEFT TURN
SHUTDOWN	x	Blink Red	x	x	x	x	x	x
MOVE IN	Ping Dis = R	Blink Yellow	MEASURE	x	x	x	x	x
LEFT TURN	x	Yellow	MEASURE	x	x	x	x	x
RIGHT TURN	x	Yellow	MEASURE	x	x	x	x	x
FORWARD	x	Yellow	MEASURE	x	x	x	x	x
MEASURE	x	Green	LOCATION DETECT	x	x	x	x	x

Diagram 4: State Table

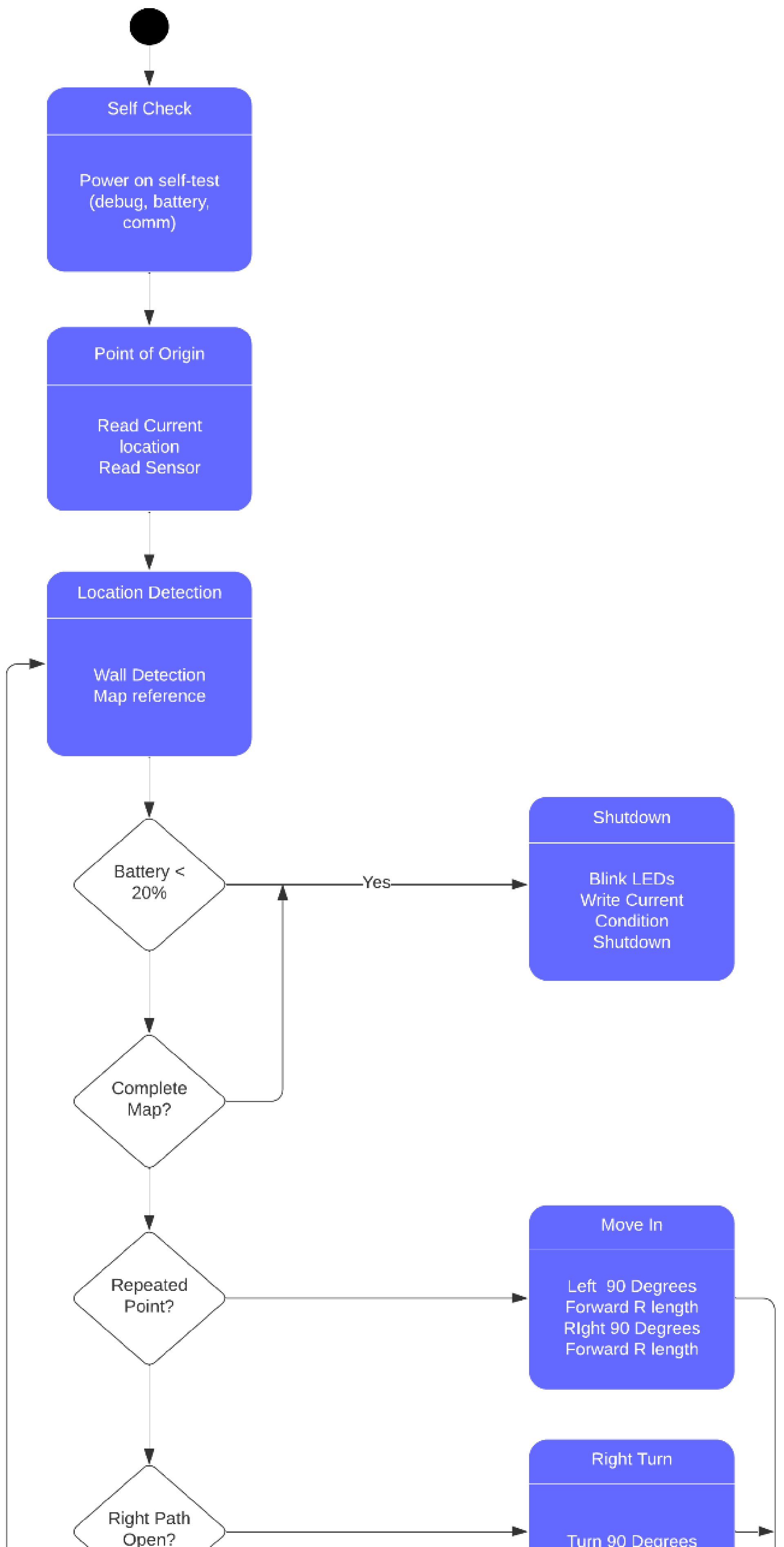
Self Check
Power on self-test (debug, battery, comm)
Point of Origin
Read Current location
Read Sensor
Location Detection
Wall Detection
Map reference
Battery < 20%
Yes
Shutdown
Blink LEDs
Write Current Condition
Shutdown
Complete Map?
Repeated Point?
Move In
Left 90 Degrees
Forward R length
RIght 90 Degrees
Forward R length
Right Path Open?
Right Turn
Turn 90 Degrees

Diagram 5: State Machine Diagram

Build Instructions

1. Build
2. Program .bin to the processor
3. Button to initiate point of origin

Future

The past can't be changed and the future doesn't exist. -Marcus Aurelius

There are a few paths to future development that I would like to concentrate on.

1. Spatial awareness improvements through
    a. Improved sensors (mechanical gyros, etc)
    b. Additional sensors (gps, LIDAR, hub unit)
2. Computational improvements through
    a. Mapping concentration values in 3D
    b. Building probable gas concentration vectors first on the backend then porting that the embedded system (possibly using the Nvidia Jetson)
3. Abstracting the application - the actual sensor/measurement doesn't matter
    a. Porting the spatial awareness aspect to a genetic lib
    b. Building the vector analysis for determining any gas, liquid or field that may emit from x number of sources and give probability of origin locations

Grading

Link to Code

Appendix

1. Meets goals
2. Completeness of deliverables
3. Clear intentions and working code
4. Reusing Code
5. Originality and scope of goals
6. Self Assessment
7. Power Analysis, FW update or system profile
8. Version control was used

References:

Board Documentation
STM32F411VE Documentation