# Task 1: Motor Driver (Mot)

## Operations:

* Internal:
  + Operate Motors
    - Convert speed/turn -> mot\_A/mot\_b
    - Ramp acceleration
  + Read Encoders
    - Convert encoder raw data -> disp/ang\_disp
  + Run Control Systems
    - Speed Controller (normal operation)
    - Position Controller (specific operations like ‘hold position’)
* Interface
  + Receive Motor State (From Nav)
  + Send Motor Status (To Nav/Safety)
  + Send Encoder Data (To Nav)

## Shares:

* (R/W) Motor Flag Register

uint8\_t mot\_flag // Flag Register for Control Booleans

* (R) Inputs from Nav

int8\_t bot\_speed // ± [ft/s] Speed (scalar) of the robot

int8\_t bot\_turn // ± [deg/s] Turning rate of the robot

* (W) Outputs to Nav

float bot\_disp // ± [ft] Displacement of the robot

int16\_t bot\_angd // ± [ deg] Angular displ of the robot

# Task 2: Geofence (Geo)

## Operations:

* Internal:
  + Read from GPS
  + Run Geofence Algorithm
* Interface
  + Send Fence Data to Nav
    - Dist/Bear

## Shares:

* (W) Outputs to Nav

float bot\_dist // ± [ft] Closest distance to fence

int16\_t bot\_bear // ± [ deg] Bearing to closest point

# Task 3: Safety Sensor Suite (Sen)

## Operations:

* Internal:
  + Read from each sensor
* Interface
  + Send Sensor Data to Nav
    - Dist/Bear

## Shares:

* (W) Outputs to Nav [For Each Sensor]

float XXX\_dist // ± [ft] Dist registered by sensor XXX

int16\_t XXX\_bear // ± [ deg] Bearing of sensor XXX

# Task 4: Navigation (Nav)

## Operations:

* Internal:
  + Process Sensor Data
  + Determine Path
* Interface
  + Control Motors
    - Speed/Turn
  + Accept User Commands (From UI Task)
  + Interact with slave µC

## Shares:

* (R/W) Motor Flag Register

uint8\_t mot\_flag // Flag Register for Control Booleans

* (R) Sensor Inputs from Mot

float bot\_disp // ± [ft] Displacement of the robot

int16\_t bot\_angd // ± [ deg] Angular displ of the robot

* (R) Sensor Inputs from Sen [For Each Sensor]

float XXX\_dist // ± [ft] Dist registered by sensor XXX

int16\_t XXX\_bear // ± [ deg] Bearing of sensor XXX

* (W) Outputs to Mot

int8\_t bot\_speed // ± [ft/s] Speed (scalar) of the robot

int8\_t bot\_turn // ± [deg/s] Turning rate of the robot

# Task 5: User Interface (UI)

Handles all User Interface related operations - reads Bluetooth Connected Vector SVG Icon - PNG Repo Free PNG Icons, manual buttons, etc. and determines appropriate course of action.

## Operations:

* Internal:
  + Process UI inputs
  + Determine proper outputs
  + Generates LCD display
* Interface
  + Operates Bluetooth Connected Vector SVG Icon - PNG Repo Free PNG Icons driver
  + Reads Manual Buttons (if enabled)
  + Outputs commands to Nav (to be routed to other tasks)
  + Controls LCD

## Shares:

* (R/W) LCD Flag Register

uint8\_t lcd\_flag // Flag Register for LCD Booleans

* (R) Potentially any known read-only share from other tasks

?? Whatever we want on the LCD that is not generated by UI ??

* (W) Outputs to Nav

~~int8\_t bot\_speed // ± [ft/s] Speed (scalar) of the robot~~

~~int8\_t bot\_turn // ± [deg/s] Turning rate of the robot~~

# Task 6: LCD Driver (LCD)

Operates regularly to ensure LCD operation is non-blocking

## Operations:

* Internal:
  + Operate LCD
* Interface
  + Receive LCD Data

## Shares:

* (R/W) LCD Flag Register

uint8\_t lcd\_flag // Flag Register for LCD Booleans

## Queues:

* (R) LCD info to be printed from UI

char lcd\_queue // Char to be printed (& other LCD codes)

# Task 7: Slave µC Interface (Com)

Interacts with slave µC to receive sensor data over \_\_(com protocol)\_\_ and send/receive commands for crowd interraction

## Operations:

* Internal:
* Interface