Nav.c

Tells the mouse where to go based on current position and maze state. Contains a few selectable algorithms (Command List Follower, Right-wall Follower, Random Walk, Flood-Fill Follower)

Maze.c

Remembers walls seen by the sensors. Stores bit flags for each wall along with flood-fill routing information in two arrays. Floods maze during spare CPU time. Contains a few utility functions to get cells, set flags, translate relative headings (front, left, ...) to absolute headings (north, east, ...), etc. Saves/loads maze from flash*.

Motion.c

Motion control state machine. Maintains a queue of movement commands (left90 , right90, fwdFollow, ...) and carries them out by incrementing the motor position set-points over time. Mainly trusts encoders, correcting and aligning to walls with sensors readings when available. Keeps track of currently occupied cell and heading. Tells maze module to perform a map and re-flood upon entering a new cell.

* todo

Motor.c

Manages hardware PWMs. Performs PI control that tries to move motors to a given encoder count.

Encoder.c

Handles encoder interrupts, keeps encoder counts, contains quadrature state machine

Adc.c

Manages ADCs for IR sensors. Pulses sensors for ambient light correction. Handles misc. inputs

Motor Driver IC

Shaft Encoders

IR Emitter + Detector

Push-buttons

Ui.c

Misc. debugging-related code.

Sched.c

Very simple, hardcoded, FIFO-like, nonpreemptive scheduler.

Blue LEDs!