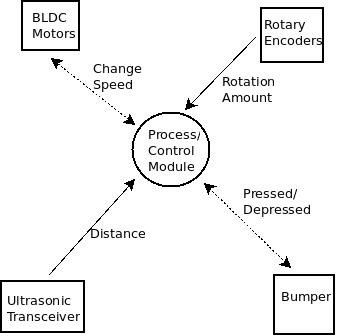
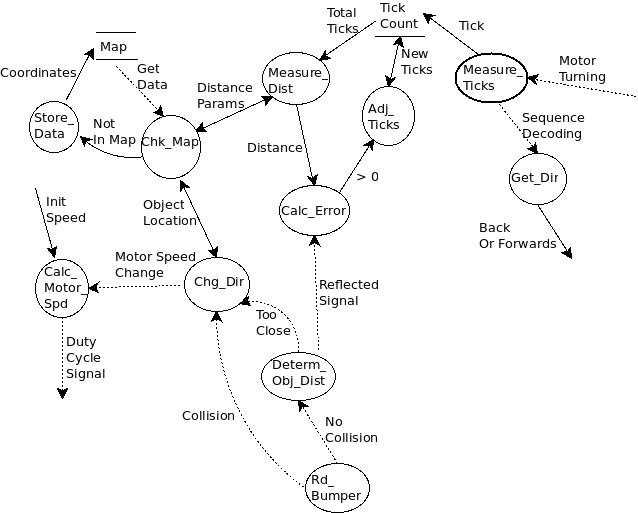
**Requirements and Architecture Model For Simplified Autonomous Two-Wheel Mobile Robot**

**By: Kerry Murphy**

****

**Data/Control**

**Context Diagram**

**Data/Control**

**Flow Diagram**

**PSPECS & CPECS (Process & Control Specifications)**

**Process/Control Module**

This process is the main control process for autonomous operation of the mobile robot. The BLDC Motors provide means of locomotion to the robot. It is sends and receives control signals from controller which control speed and determine position of the motor. The Rotary Encoder signals the position and rate of rotation of the wheels whereby distance traveled can be calculated. The Ultrasonic Transceiver is used to determine location of robot with respect to an object (wall, chair, tree, etc.) in the environment. Bumpers send signals to the controller to determine when the robot collides with an object in the environment.

**Measure\_Dist (Software)**

This process translates the total ticks in the counter store into a distance unit which indicates how far the robot has traveled from a reference point. It also stores the distance value in the map. The output is sent for compare with the information gained from the Rotary Encoder.

**Measure\_Ticks (Software and Digital Small Signal)**

This process takes the output from the Rotary Encoder signal and converts that to a direction of rotation and a rotation unit (i.e. 1 tick) which is stored in the Tick\_Count data store.

**Store\_Data (Software and Flash)**

This process takes data not stored in Map and recreates it in the Map.

**Chk\_Map (Software and Flash)**

This process gets locality map data from Map store, distance parameters, and object location from near object collision. It determines if this information is already in the map. If it is, then it sends info back to other processes for updating distances and turning radius. If not in the map, then it sends location data to the Store\_Data process.

**Adj\_Ticks (Software)**

This process is responsible for taking error and current tick count as input and outputting adjusted ticks to be stored in Tick Count store.

**Get\_Dir (Software and Digital Small Signal)**

This process takes the Sequence Decoding from the Rotary Encoder and calculates a direction-either forward or backward.

**Calc\_Error (Software and Analog Low Power Small Signal Low Frequency)**

This process gets the distance unit and compares it with received sound wave signal to determine if error in distance from reference point.

**Calc\_Motor\_Spd (Software and Digital High Power High Frequency)**

This process determines appropriate level of direct current voltage to send to each motor

**Chg\_Dir (Software and Flash Digital Low Power)**

This process accepts three inputs and can generate two outputs. It can take object location information from the map store and predict future turns. It is also notified if a collision occurs or an object is too close. The updated location info is sent to the map and used to change the speed of the motors.

**Determ\_Obj\_Dist (Software Digital Low Power and/or Analog Low Power Small Signal Low Frequency)**

This process determines that no collision has occurred and therefore output if too close to an object or far away.

**Rd\_Bumper (Digital Low Power)**

This process reads the pressure sensor and determines if robot has collided with an object or not.

**TSPEC (Timing Specification)**

Output PWM frequency to the BLDC motors must occur at minimum 5 kHz or 200 microseconds. External sensors such as Ultrasonic sensors and bumper must be accurately read within 1 millisecond. Rotary Encoder must accurately be read within 100 milliseconds. Map locality info must be read and updated within 1 millisecond.

**Requirements Dictionary**

**BLDC Motor**

Brush-less Direct Current motor

**Bumper**

A switch in form of a button or feelers which can be in two states- pressed or depressed

**Duty Cycle**

The percentage amount a given voltage is at a high voltage compared to low voltage for a given period

**Map**

Represents physical objects relative to each other in digital form

**PWM**

Pulse Width Modulation or a way of modulating the signal to control such things as speed

**PWM Frequency**

Specifies the rate with respect to time at which the modulated signal is generated

**Reflected Signal**

A signal such as a sound wave or radio wave which is reflected from a collision into an object

**Rotary Encoder**

An electro-mechanical device which converts angular motion into an analog or digital voltage

**Sequence Decoding**

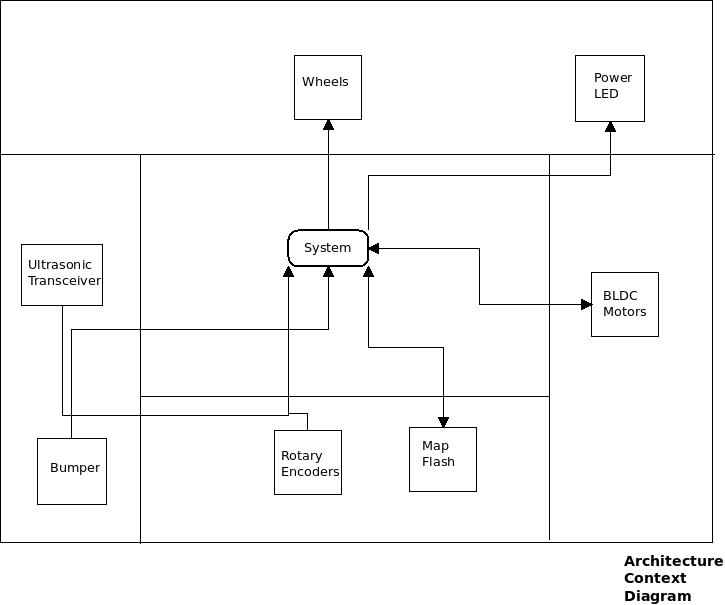
A modulated signal is compared with a signal on another channel or a signal pattern in memory to determine things like rate of change or direction

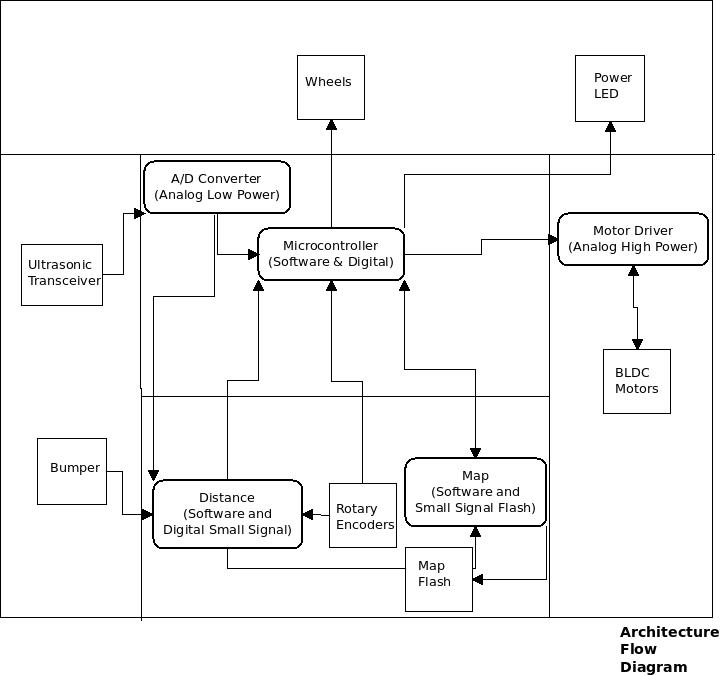
**Tick**

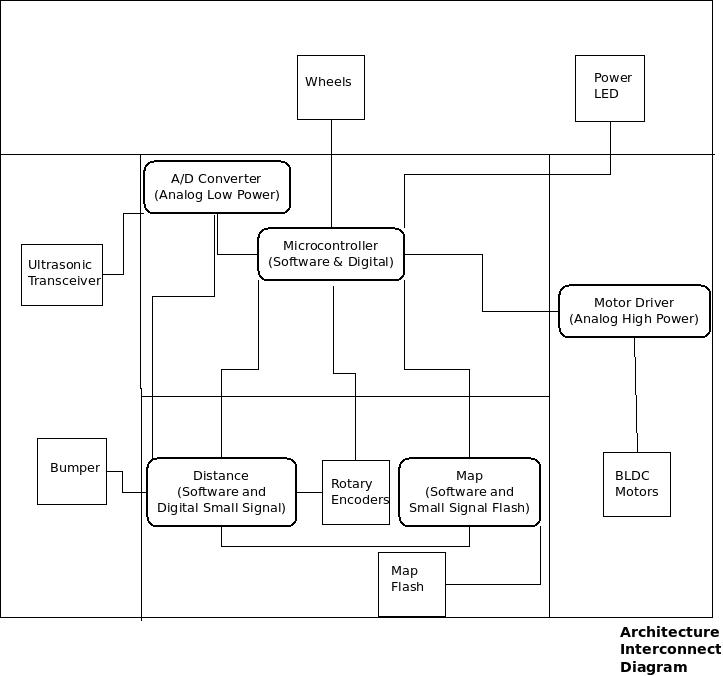
One unit representing an actual change in voltage transition (low-to-high or high-to-low) incoming from Rotary Encoder

**Ultrasonic Transceiver**

An electronic device which generates inaudible high frequency sound waves and can also detect the reflection of said sound wave







**Architecture Module Specification (AMS)**

**Microcontroller**

This module is responsible calculating each motor speed based on distance (digital input) and direction and map feedback. It is also responsible for updating and configuring the Map module in software.

**Distance**

This module is responsible for calculating distance traveled and distance to an object. It takes a digital input from bumper to determine collision. It also takes as feedback from Map module distance for redundancy, self-testing checking.

**Map**

This module is responsible for keeping map information of environment obstacles stored in memory on the microcontroller as well as majority of information stored in external flash and making it easily accessible to other modules.

**Motor Driver**

This module is responsible taking control signal from microcontroller and using de-coupled power supply to provide high current to motor and protect microcontroller.

**A/D Converter**

This module is a discrete component not implemented in software. It could be part of the microcontroller, but is shown separately to demonstrate an analog signal needs to be converted to digital to be compatible with inputs of other modules.

**Architecture Interconnect Specification**

All modules are connected via software with the exception of A/D Converter, Map, and Motor Driver. A/D Converter uses a small wire anywhere between 28 to 33 gauge. Map uses similar specification to connect to flash but with more than one wire. The microcontroller connects to Motor Driver using 28 to 33 gauge wire. Motor Driver uses 18 to 22 gauge wire to connect to BLDC motors.

**Timing Requirements**

The Distance, Map, and A/D Converter modules need to respond within 1 millisecond. The microcontroller module must drive BLDC motors at 200 microsecond intervals. Therefore, Motor Driver module must meet 200 microsecond requirement.