Lab 3 - Pre-Lab & Pseudo Code
Alex Gall, Noah Terry

Introduction

This lab will allow us to monitor the spin of the motors by implementing functionality to read the left and right encoders. This lab will also allow us to monitor the charge of the batteries and prevent draining the batteries past the minimum voltage.

Pseudocode

Functions in Encoder.h/c

FUNCTION Encoders_Init()
DESCRIPTION Initializes the encoders
INPUTS NONE
RETURNS NONE

//Need to enable global interrupts…?
//sei() also does this? Do we need to call sei twice?
SET the 'SREG' register bit 'I' TO 1 //Bit 7

// Enable interrupts on the INT6 pin
SET the 'EIMSK' register bit 'INT6' TO 0 // (needs to be cleared before setting EICRB)
SET the 'EICRB' register bit 'ISC60' TO 1 // Bit 4
SET the 'EIMSK' register bit 'INT6' TO 1 // Bit 6

// Enable interrupts on the PCINT4 pin
SET the 'PCICR' register bit 'PCIE0' TO 1 // Bit 0
SET the 'PCMSK0' register bit 'PCINT4' TO 1 // Bit 4

END FUNCTION

FUNCTION Counts_Left()
DESCRIPTION Returns the number of counts from the left encoder
INPUTS NONE
RETURNS Number of counts from the left encoder

RETURN _left_counts

END FUNCTION

FUNCTION Counts_Right()

DESCRIPTION Returns the number of counts from the right encoder
INPUTS NONE
RETURNS Number of counts from the right encoder

RETURN _right_counts

END FUNCTION

FUNCTION Rad_Left()

DESCRIPTION Returns the number of radians from the left encoder
INPUTS NONE
RETURNS Number of radians for the left encoder

CALL Counts_Left()
CONVERT Counts_Left() to radians
RETURN radian conversion

END FUNCTION

FUNCTION Rad_Right()

DESCRIPTION Returns the number of radians from the right encoder
INPUTS NONE
RETURNS Number of radians for the right encoder

CALL Counts_Right()
CONVERT Counts_Right() to radians
RETURN radian conversion

END FUNCTION

Functions in Battery_Monitor.h/c

FUNCTION Battery_Monitor_Init()

DESCRIPTION Initializes functionality for battery monitoring
INPUTS NONE
RETURNS NONE

// Need to set the conversion frequency to be between 50kHz and 200kHz
// Our processor is at 16MHz -> dividing by 128 gives us 125kHz
SET the register 'ADCSRA' bit 'ADPS0' TO 1 // Bit 0
SET the register 'ADCSRA' bit 'ADPS1' TO 1 // Bit 1
SET the register 'ADCSRA' bit 'ADPS2' TO 1 // Bit 2

// Need to set the correct channel to read from and the gain value
// Channel: ADC6. Gain: 1x
SET the register 'ADMUX' bit 'MUX1' TO 1 // Bit 1
SET the register 'ADMUX' bit 'MUX2' TO 1 // Bit 2
SET the register 'ADMUX' bit 'MUX4' TO 1 // Bit 4

// Enable the car's analog reference & decoupling capacitor
// I.e. enable external reference and capacitor
SET the register 'ADMUX' bit 'REFS0' TO 1 // Bit 6

//Enable the ADC
SET the register 'ADCSRA' bit 'ADEN' TO 1 // Bit 7

//Enable ADC conversion complete interrupt
SET the register 'ADCSRA' bit 'ADIE' TO 1 // Bit 3

//Disable digital inputs to enable the ADC on the ADC6
SET the register 'DIDR0' bit 'ADC6D' TO 1 // Bit 6

END FUNCTION

FUNCTION Battery_Voltage()

DESCRIPTION Measures the voltage of the battery using the A/D measurement
INPUTS NONE
RETURNS Battery voltage

START analog to digital conversion // ADC SRA register bit ADSC

READ data in the A/D low register // register ADCL bit ADC6
READ data in the A/D high register // register ADCH bit ADC6

COMBINE data from the high and low register and store in variable v_bat
RETURN v_bat

END FUNCTION

Planned Schedule and Task Assignment

We will start by pair programming on Tuesday and continue working Thursday and Friday. We plan on completing the code on Friday but will use Saturday for debugging if necessary. We plan on completing the demo and report on Saturday.