**PWM.c / PWM.h**

This module is responsible for the control of the Pulse Wave Modulation output of the atmega328 processor to the H-bridges which drives the motor. The Code utilizes the two 8-bits counter/timer on the atmega328 and outputs 4 PWM square waves. The mode of the counter/timer have been specified within the header file. The Duty cycle of the four individual square waves is modifiable via this function:

Void XX\_setPWM(uint8\_t value);

The XX represent the input pins on the h-bridges at which the corresponding PWM will be outputted to. (A1, A2, B1, B2) . The corresponding output port will be specified in the header file.

Eg. A1\_setPWM(0x7F); Will set the duty cycle of PWM on A1 to 50%.

When using the function from this code, make sure to call the init\_PWM to initialize the counters/timer mode before calling the actual functions.

**motor.c / motor.h**

This module serve as the driver for the motors, it is an abstraction layer of the PWM module providing interfaces between the PWM output and the actual motor behavior. This module provide functions which controls the behavior on each individual motor by modifying the PWM outputs to the h-bridges, as well as general control such as forward and break. Before calling the functions within this module, make sure to call the init\_motor(), which initialize the PWM setting and set both motors to brake.

**Sensor.c / Sensor..h**

This module provides accesses to the HC-HR04 ultrasonic sensor. The module contain two function, which query corresponding sensor to detect the distance between the obstacle and the sensor. The sensors will only be active when the

uint8\_t find\_distance\_XXXX();

function is called, replace XXXX with the direction at which the Sensor is facing (Left and Right). The function call will return an unsigned 8 bit number(0-255) indicating the distance from the object that the sensor detects. Make sure to call the init\_Sensor function to initialize the timer and DDR before using the functions.

When the sensor receives a trigger, which is a short pulse, from the processor, it will sends out an ultrasonic wave and set its ‘echo’ signal to high. When the Sensor picked up the actual echo of its ultrasonic wave that bounced off of certain obstacle, it will set the echo signal to low. When the find\_distance function is called, the processor will send out the trigger pulse to the Sensor and wait for the echo signal from the sensor to go high. The echo signals of the sensors are wired to the two external interrupt pin on the processor, INT0 and PCINT2. The program is configured to fire interrupts on a logic change in these two pins, which means an interrupt will fire on both the rising\_edge and the falling\_edge of the external interrupt pins.

The interrupt service routine is expected to be invoked twice during the sensor operation. First invoke will reset the internal 16-bit counter of the processor to initiate the timer. The second invoke will retrieve the timer value and convert this value into distance in centimeter and save the result in a global distance variable. The function will then return this distance variable. The flow of this interrupt service routine is controlled by a flag.

The function will wait for 60 milliseconds for the sensor to complete its operation using the \_delay\_ms() function. After the delay is over, the function will return the value of the distance variable regardless if it was correctly updated or not.