Andrew Capatina
11/4/2017
Version 1.0

NEED TO UPDATE ALGORITHM FOR CALLBACK FUNCTIONS

Low Level Speed Conversion Algorithm

Setup:

```
1. Take note of timer handle tim1.
2. Initialize function prototype with return type integer conversion_mode(). Function takes no arguments.
3. Initialize bool variable start_flag.
4. Initialize bool variable mode_state.
5. Initialize bool interrupt_flag.
6. Initialize array of size 5 to_average.
7. Call HAL_TIM_Base_Start and pass in handle tim1 by reference.
```

Main:

```
1. Initialize uint8_t variable mode.
2. Initialize integer count to zero.
3. Set wheel_radius to return value of get_wheel_size.
4. Set mode to zero.
5. Stay in loop until start_flag changes to true.
6. While mode is zero:
    a. While count is not equal to five:
        i. Set speed equal to return value of conversion_mode(). Pass in count by value.
        ii. Assign speed to index of to_average.
        iii. If speed isn't
        iv. If count is equal to five, sum all array contents of to_average and divide by number of members.
    b. Write to LCD function that displays speed. Pass in speed as function parameter.
        i. Will be implemented at a later time.
7. Go to mode 1
    a. Will be implemented at a later time.
```

conversion_mode(uint8_t index)

```
1. Initialize integer return variable speed to -1.
2. Initialize integer variable count.
3. Initialize bool variable detected.
4. Insert do while. Conditional will be to wait until detected is equal to zero.
    a. If detected is equal to zero, do the following:
        i. Call HAL_TIM_IC_Start(). Pass handle to TIM1 by reference and pass in the first timing channel.
        ii. While speed is equal to -1:
            1. Set count equal to __HAL_TIM_GET_COUNTER(). Pass in handle to TIM1 as reference.
```

b. Call HAL_TIM_IC_Stop() and pass in TIM1 handle as reference as well as timer channel 1 by value.
c. Apply conversion algorithm formula, assign value to speed.
d. Return speed.