Quick Reference Guide to the KIPR Wallaby(Updated 2/1/17)

```
printf("text\n");                                  // Prints the specified text to the screen
wait_for_milliseconds(# milliseconds);             // Waits specified number of milliseconds before next line
msleep(# milliseconds);                            // Another name for wait_for_milliseconds (identical)
motor(port #, % velocity);                         // Turns on motor with specified port # at % velocity
motor_power(port #, % power);                      // Turns on motor with specified port # at % power
mav(port #, velocity);                             // Move motor at specified velocity (# ticks per second)
mrp(port #, velocity, position);                   // Move motor to specified relative position (in # ticks)
ao();                                              // All off; turns all motor ports off
enable_servos();                                   // Turns on servo ports
disable_servos();                                  // Turns off servo ports
set_servo_position(port #, position);              // Moves servo in specified port # to specified position
wait_for_light(port #);                            // Waits for light in specified port # before next line
wait_for_touch(port #);                            // Waits for touch in specified port # before next line
analog(port #);                                    // Get a sensor reading from a specified analog port #
digital(port #);                                   // Get a sensor reading from a specified digital port #
shut_down_in(time in seconds);                     // Shuts down all motors after specified # of seconds
```

```
camera_open();                                     // Opens the camera for use
camera_open_black();                               // Opens the black camera for use
camera_close();                                    // Closes the current camera instance
camera_update();                                   // Pulls a new image from the camera for processing
get_object_center_x(channel #, object #);          // The x-axis center of a specified object on a specified channel
get_object_count(channel #);                       // Counts the number of objects using the given channel
```

```
create_connect();                                  // Establishes a connection to the create
create_disconnect();                               // Disconnects from the create
create_drive_direct(l_speed, r_speed);             // Moves left(l) and right(r) create motors at specified speeds
create_stop();                                     // Turns all create motors off
get_create_total_angle(angle);                     // Gets the creates current angle; negative is counterclockwise
set_create_total_angle();                          // sets the total angle of the create to the specified value
```