Threading

KISS Institute for Practical Robotics © 1993 – 2025 KIPR

KISS Institute for Practical Robotics © 1993 – 2025 KIPR

Introduction

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

- Threading allows a program to do multiple things at once
- In this section, we will demonstrate how to use threading to run two functions at the same time

KISS Institute for Practical Robotics © 1993 – 2025 KIPR

Function 1

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Source Code

```
1 void spin(int seconds)
2 {
3     motor(0, 100);
4     motor(1, 100);
5     msleep(1000 * seconds);
6     ao();
7 }
```

This is a simple function which causes the robot to spin in place for the number of seconds specified by the seconds parameter

KISS Institute for Practical Robotics © 1993 – 2025 KIPR

Function 2

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Source Code

```
1  void servo_to_top()
2  {
3      int target_pos = 2000;
4      int curr_pos = get_servo_position(0);
5      int pos_diff = target_pos - curr_pos;
6
7      /* 50 iterations * 100 ms / iteration = 5 seconds */
8      int iters = 50;
9      int interval = pos_diff / iters;
10
11     for ( ; curr_pos < target_pos; curr_pos += interval) {
12         set_servo_position(0, curr_pos);
13         msleep(100);
14     }
15 }
```

This function uses the current position of servo 0 and the target position to move it to the target position in 5 seconds.

KISS Institute for Practical Robotics © 1993 – 2025 KIPR

Example main()

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Source Code

```
1  #include <kipr/wombat.h>
2
3  int main()
4  {
5      enable_servos();
6      set_servo_position(0, 0);
7      msleep(1000);
8
9      thread servo_thread;
10     servo_thread = thread_create(servo_to_top);
11     thread_start(servo_thread);
12
13     spin(3);
14     thread_wait(servo_thread);
15     thread_destroy(servo_thread);
16     disable_servos();
17     return 0;
18 }
```

Explanation on following slides

KISS Institute for Practical Robotics © 1993 – 2025 KIPR

Example main(): Explanation

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Source Code

```
1 enable_servos();
2 set_servo_position(0, 0);
3 sleep(1000);
```

This code should already be familiar to you. Here we are just resetting the servo position for the purposes of demonstration.

Note: Enabling and disabling servos in one thread enables or disables them in all threads

KISS Institute for Practical Robotics © 1993 – 2025 KIPR

Example main(): Explanation

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Source Code

```
1 thread servo_thread;
2 servo_thread = thread_create(servo_to_top);
3 thread_start(servo_thread);
```

To keep track of our new thread we use a special type called `thread`.

Then, we tell it that when it starts, it should run the `servo_to_top()` function.

Finally, we actually start the new thread.

Now we have the `servo_to_top()` function running in a new thread!

KISS Institute for Practical Robotics © 1993 – 2025 KIPR

Example main(): Explanation

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Source Code

```
1 drive(3);
2 thread_wait(servo_thread);
3 thread_destroy(servo_thread);
```

Now we call our `drive()` function, and we are running `drive()` and `servo_to_top()` simultaneously!

`thread_wait()` tells our program to wait until `servo_thread` is done before continuing. If we didn't have this, the program would end before the `servo_to_top()` finished, because it runs for 5 seconds, while our `drive()` call only runs for 3.

Finally, always run `thread_destroy()` to clean up the thread when you are done with it

KISS Institute for Practical Robotics © 1993 – 2025 KIPR

Note: Function Parameters

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

- You may have noticed that there is no way to pass parameters to a threaded function. This is a limitation of the KIPR threading library.
- You can use global variables to pass values between functions, or, if you know how, you can use the `pthread` library
- However, for most situations the KIPR threading library should be sufficient

KISS Institute for Practical Robotics © 1993 – 2025 KIPR