wait_for_light and shut_down_in

Botball Tournament Functions

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

These two functions should be two of the first lines of code in your Botball tournament program!

```
wait_for_light(0);
// Waits for the light on port #0 before going to the next line.

shut_down_in(119);
// Shuts down all motors after 119 seconds (just less than 2 minutes).
```

- **This function call should come immediately after the `wait_for_light()`** in your code.
- If you do not have this function in your code, your robot may not automatically turn off its motors at the end of the Botball round and **you will be disqualified**!

Tournament Templates

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Source Code

```c
#include <kipr/wombat.h>

int main()
{
    // initial variable declarations, camera and servo may go here
    wait_for_light(0); // change the port number to match the port you use
    shut_down_in(119); // shut off the motors and stop the robot after 119 seconds

    // This is where most of your code will go

    // Specifically the code to play the game
    // after the light comes on (after hands off)
    return 0;
}
```

Running a Botball Tournament Function

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Description: Write a program that waits for a light to come on, shuts down the program in 5 seconds, drives the DemoBot forward until it detects a touch, and then stops.

Analysis: What is the program supposed to do?

Pseudocode	Comments
1. Wait for light.	// 1. Wait for light.
2. Shut down in 5 seconds.	// 2. Shut down in 5 seconds.
3. Drive forward.	// 3. Drive forward.
4. Wait for touch.	// 4. Wait for touch.
5. Stop motors.	// 5. Stop motors.
6. End the program.	// 6. End the program.

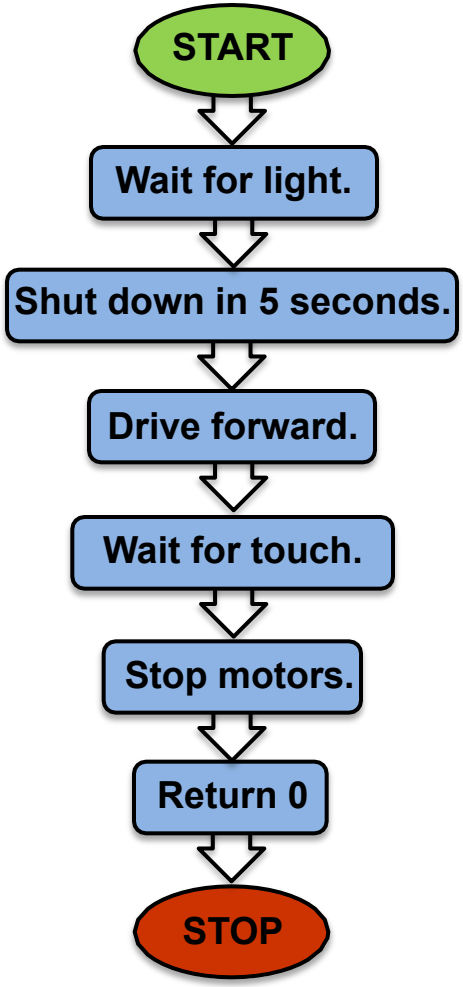
Running a Botball Tournament Program

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Analysis:

Flowchart



Running a Botball Tournament Program

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Solution:

Pseudocode

```
1.  Wait for light.
2.  Shut down in 5 seconds.
3.  Drive forward.
4.  Wait for touch.
5.  Stop motors.
6.  End the program.
```

Source Code

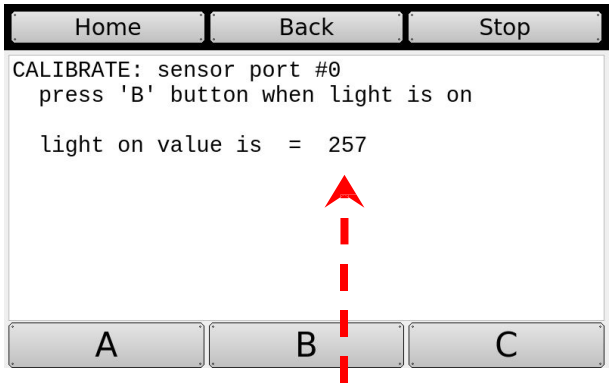
```
1  #include <kipr/wombat.h>
2
3  int main()
4  {
5      wait_for_light(0);
6
7      shut_down_in(5);
8
9      while (digital(0) == 0)
10     {
11         motor(0, 100);
12         motor(3, 100);
13     }
14     ao();
15     return 0;
16 }
```

Execution:

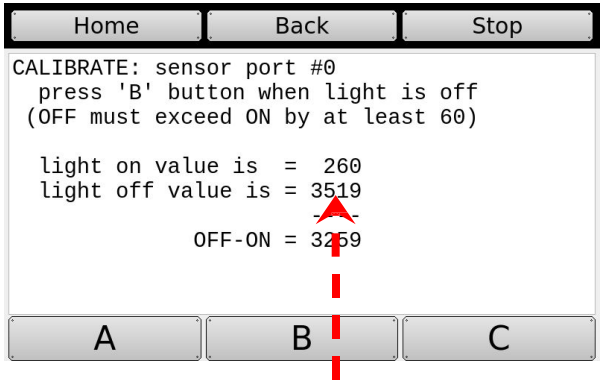
Compile and run your program (test it at different distances).

wait_for_light Calibration Routine

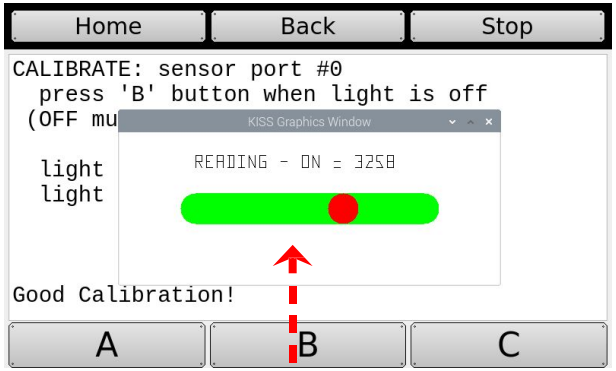
When you use the **wait_for_light ( )** function in your program, the following calibration routine will run automatically.



When the light is *on* (low value), press the "B" button.



When the light is *off* (high value), press the "B" button.



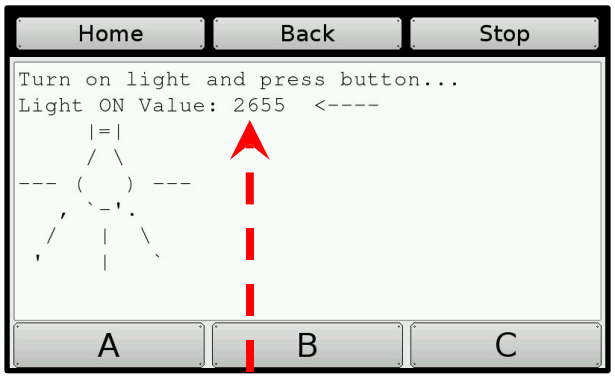
You will get a "Green Bar - Reading On" when done *correctly*.

You will get a "BAD CALIBRATION" message when not done correctly, and you will need run through the routine again.

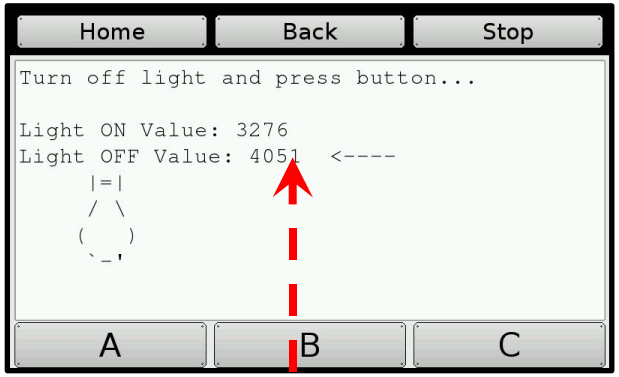
Note: For Botball, **wait_for_light ( )** should be one of the first functions called in your program.

wait_for_light Calibration Routine

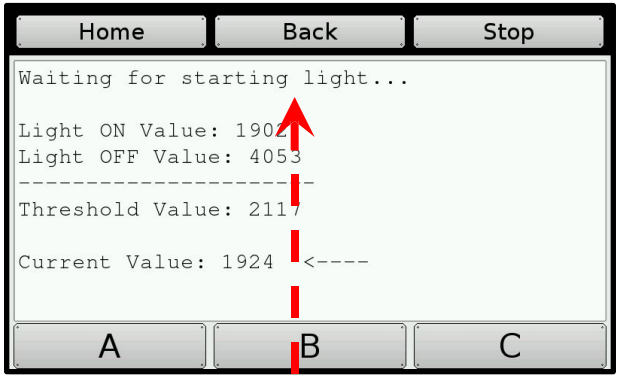
When you use the **wait_for_light ( )** function in your program, the following calibration routine will run automatically.



When the light is *on* (low value), press the "push" button.



When the light is *off* (high value), press the "push" button.



You will get a "Waiting for starting light" when done *correctly*. You will get a "BAD CALIBRATION" message when not done correctly, and you will need to push the "push" button to run through the routine again.

"push" button

Note: For Botball, **wait_for_light ( )** should be one of the first functions called in your program.

Conclusion

KISS INSTITUTE FOR PRACTICAL ROBOTICS

Botball®

Reflection:

- What happens if the touch sensor is pressed in ***less than 5 seconds*** after starting the program?
- What happens if the touch sensor is **not** pressed in ***less than 5 seconds*** after starting the program?
- What is the best way to guarantee that your program will ***start with the light*** in a Botball tournament round? (**Answer:** `wait_for_light(0)`)
- What is the best way to guarantee that your program will ***stop within 120 seconds*** in a Botball tournament round? (**Answer:** `shut_down_in(119)`)

Use these functions in your Botball tournament code!