

# Reference Manual

Generated by Doxygen 1.7.3

Mon Feb 21 2011 22:00:41



# Contents

<b>1</b>	<b>File Index</b>	<b>1</b>
1.1	File List . . . . .	1
<b>2</b>	<b>File Documentation</b>	<b>3</b>
2.1	defines.h File Reference . . . . .	3
2.1.1	Define Documentation . . . . .	4
2.1.1.1	ALTER . . . . .	4
2.1.1.2	BLACK . . . . .	4
2.1.1.3	BLUE . . . . .	4
2.1.1.4	CALIBRATE . . . . .	4
2.1.1.5	CORNER . . . . .	4
2.1.1.6	EAST . . . . .	4
2.1.1.7	FINISHED . . . . .	4
2.1.1.8	FOLLOW . . . . .	4
2.1.1.9	FOLLOW_LEFT . . . . .	4
2.1.1.10	FOLLOW_LINE . . . . .	4
2.1.1.11	FOLLOW_RIGHT . . . . .	4
2.1.1.12	FORWARD . . . . .	4
2.1.1.13	GREEN . . . . .	4
2.1.1.14	LEFT . . . . .	4
2.1.1.15	LIGHT_BLUE . . . . .	4
2.1.1.16	LIGHT_PORT . . . . .	4
2.1.1.17	MOTOR_LEFT . . . . .	4
2.1.1.18	MOTOR_RIGHT . . . . .	5
2.1.1.19	NORTH . . . . .	5
2.1.1.20	NUM_COLORS . . . . .	5
2.1.1.21	PAUSED . . . . .	5
2.1.1.22	RAND_MAX . . . . .	5
2.1.1.23	RED . . . . .	5
2.1.1.24	RIGHT . . . . .	5
2.1.1.25	SEARCH . . . . .	5
2.1.1.26	SEARCH_CORNER . . . . .	5
2.1.1.27	SOUTH . . . . .	5
2.1.1.28	START . . . . .	5
2.1.1.29	SWITCH_WAIT . . . . .	5
2.1.1.30	TEST . . . . .	5
2.1.1.31	TURN . . . . .	5
2.1.1.32	UNKNOWN . . . . .	5
2.1.1.33	UNSURE . . . . .	5

2.1.1.34	WEST	5
2.1.1.35	WHITE	5
2.1.1.36	YELLOW	5
2.2	functions.c File Reference	5
2.2.1	Function Documentation	6
2.2.1.1	change_position	6
2.2.1.2	check_color	6
2.2.1.3	query_turn	6
2.2.1.4	robot_follow	7
2.2.1.5	set_both_motor_speed	7
2.2.1.6	set_motor_threshold	7
2.3	functions.h File Reference	7
2.3.1	Function Documentation	8
2.3.1.1	change_position	8
2.3.1.2	check_color	9
2.3.1.3	query_turn	9
2.3.1.4	robot_follow	9
2.3.1.5	set_both_motor_speed	9
2.3.1.6	set_motor_threshold	10
2.3.2	Variable Documentation	10
2.3.2.1	checked	10
2.3.2.2	color	10
2.3.2.3	colorCal	10
2.3.2.4	make_turn	10
2.3.2.5	map	10
2.3.2.6	motor_left	10
2.3.2.7	motor_right	10
2.3.2.8	old_color	10
2.3.2.9	orientation	10
2.3.2.10	x_position	10
2.3.2.11	y_position	10
2.4	kernel_cfg.c File Reference	10
2.4.1	Define Documentation	14
2.4.1.1	__STK_UNIT	14
2.4.1.2	__TCOUNT_STK_UNIT	14
2.4.1.3	IPL_MAXISR2	14
2.4.1.4	TNUM_ALARM	14
2.4.1.5	TNUM_COUNTER	14
2.4.1.6	TNUM_EXTTASK	14
2.4.1.7	TNUM_ISR2	14
2.4.1.8	TNUM_RESOURCE	14
2.4.1.9	TNUM_TASK	14
2.4.2	Function Documentation	14
2.4.2.1	_activate_alarm_cyclic_alarm_ManageState	14
2.4.2.2	_activate_alarm_cyclic_alarm_ReadButtons	14
2.4.2.3	_activate_alarm_cyclic_alarm_ReadColour	14
2.4.2.4	DeclareTask	14
2.4.2.5	DeclareTask	14
2.4.2.6	DeclareTask	14
2.4.2.7	DEFINE_CTXB	14

2.4.2.8	ManageState	14
2.4.2.9	object_initialize	14
2.4.2.10	ReadButtons	14
2.4.2.11	ReadColour	14
2.4.3	Variable Documentation	14
2.4.3.1	_stack_ManageState	14
2.4.3.2	_stack_ReadButtons	14
2.4.3.3	_stack_ReadColour	14
2.4.3.4	almcb_almval	14
2.4.3.5	almcb_cycle	14
2.4.3.6	almcb_next	14
2.4.3.7	almcb_prev	14
2.4.3.8	alminib_almval	14
2.4.3.9	alminib_autosta	14
2.4.3.10	alminib_cback	14
2.4.3.11	alminib_cntid	14
2.4.3.12	alminib_cycle	14
2.4.3.13	cntcb_almque	14
2.4.3.14	cntcb_curval	14
2.4.3.15	cntinib_maxval	14
2.4.3.16	cntinib_maxval2	14
2.4.3.17	cntinib_mincyc	14
2.4.3.18	cntinib_tickbase	14
2.4.3.19	cyclic_alarm_ManageState	14
2.4.3.20	cyclic_alarm_ReadButtons	14
2.4.3.21	cyclic_alarm_ReadColour	14
2.4.3.22	ipl_maxisr2	14
2.4.3.23	isrcb_lastres	14
2.4.3.24	isrinib_intpri	14
2.4.3.25	ManageState	14
2.4.3.26	ReadButtons	14
2.4.3.27	ReadColour	14
2.4.3.28	rescb_prevpri	14
2.4.3.29	rescb_prevres	14
2.4.3.30	resinib_ceilpri	14
2.4.3.31	SysTimerCnt	14
2.4.3.32	tcb_actcnt	14
2.4.3.33	tcb_curevt	14
2.4.3.34	tcb_curpri	14
2.4.3.35	tcb_lastres	14
2.4.3.36	tcb_next	14
2.4.3.37	tcb_tstat	14
2.4.3.38	tcb_waievt	14
2.4.3.39	tinib_autoact	14
2.4.3.40	tinib_exepr	14
2.4.3.41	tinib_inipri	14
2.4.3.42	tinib_maxact	14
2.4.3.43	tinib_stk	14
2.4.3.44	tinib_stksz	14
2.4.3.45	tinib_task	14

	2.4.3.46	<code>tnum_alarm</code>	14
	2.4.3.47	<code>tnum_counter</code>	14
	2.4.3.48	<code>tnum_exttask</code>	14
	2.4.3.49	<code>tnum_isr2</code>	14
	2.4.3.50	<code>tnum_resource</code>	14
	2.4.3.51	<code>tnum_task</code>	14
2.5	<code>kernel_id.h</code> File Reference		14
	2.5.1	Define Documentation	15
		2.5.1.1 <code>appmodel</code>	15
2.6	<code>lineFollower.c</code> File Reference		15
	2.6.1	Detailed Description	15
	2.6.2	Function Documentation	16
		2.6.2.1 <code>DeclareCounter</code>	16
		2.6.2.2 <code>DeclareTask</code>	16
		2.6.2.3 <code>DeclareTask</code>	16
		2.6.2.4 <code>DeclareTask</code>	16
		2.6.2.5 <code>ecrobot_device_initialize</code>	16
		2.6.2.6 <code>ecrobot_device_terminate</code>	16
		2.6.2.7 <code>TASK</code>	16
		2.6.2.8 <code>TASK</code>	16
		2.6.2.9 <code>TASK</code>	16
		2.6.2.10 <code>user_lms_isr_type2</code>	16
2.7	<code>lineFollower.h</code> File Reference		16
	2.7.1	Variable Documentation	17
		2.7.1.1 <code>black_count</code>	17
		2.7.1.2 <code>checked</code>	17
		2.7.1.3 <code>color</code>	17
		2.7.1.4 <code>color_cal_state</code>	17
		2.7.1.5 <code>colorCal</code>	17
		2.7.1.6 <code>enter_button</code>	17
		2.7.1.7 <code>line_found</code>	17
		2.7.1.8 <code>line_side</code>	17
		2.7.1.9 <code>make_turn</code>	17
		2.7.1.10 <code>old_color</code>	17
		2.7.1.11 <code>random</code>	17
		2.7.1.12 <code>run_button</code>	17
		2.7.1.13 <code>search_time</code>	17
		2.7.1.14 <code>state</code>	17

# Chapter 1

## File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">defines.h</a>	3
<a href="#">functions.c</a>	5
<a href="#">functions.h</a>	7
<a href="#">kernel_cfg.c</a>	10
<a href="#">kernel_id.h</a>	14
<a href="#">lineFollower.c</a> (Main code for an Autonomous Colour Line Following Robot with Area Mapping and Decision Making Abilities )	15
<a href="#">lineFollower.h</a>	16





## Chapter 2

# File Documentation

### 2.1 defines.h File Reference

#### Defines

- #define [MOTOR\\_LEFT](#) NXT\_PORT\_A
- #define [MOTOR\\_RIGHT](#) NXT\_PORT\_C
- #define [LIGHT\\_PORT](#) NXT\_PORT\_S1
- #define [BLACK](#) (0)
- #define [WHITE](#) (1)
- #define [YELLOW](#) (2)
- #define [RED](#) (3)
- #define [GREEN](#) (4)
- #define [BLUE](#) (5)
- #define [LIGHT\\_BLUE](#) (6)
- #define [UNKNOWN](#) (99)
- #define [NUM\\_COLORS](#) (7)
- #define [CORNER](#) (6)
- #define [FORWARD](#) (5)
- #define [LEFT](#) (4)
- #define [RIGHT](#) (3)
- #define [FOLLOW](#) (2)
- #define [SEARCH](#) (1)
- #define [UNSURE](#) (0)
- #define [START](#) (0)
- #define [FOLLOW\\_LINE](#) (1)
- #define [CALIBRATE](#) (2)
- #define [PAUSED](#) (3)
- #define [SEARCH\\_CORNER](#) (4)
- #define [TURN](#) (5)
- #define [FINISHED](#) (6)
- #define [FOLLOW\\_RIGHT](#) (0)

- #define [FOLLOW\\_LEFT](#) (1)
- #define [NORTH](#) (0)
- #define [EAST](#) (1)
- #define [SOUTH](#) (2)
- #define [WEST](#) (3)
- #define [TEST](#) (0)
- #define [ALTER](#) (1)
- #define [RAND\\_MAX](#) (3);
- #define [SWITCH\\_WAIT](#) (3)

### 2.1.1 Define Documentation

2.1.1.1 #define [ALTER](#) (1)

2.1.1.2 #define [BLACK](#) (0)

2.1.1.3 #define [BLUE](#) (5)

2.1.1.4 #define [CALIBRATE](#) (2)

2.1.1.5 #define [CORNER](#) (6)

2.1.1.6 #define [EAST](#) (1)

2.1.1.7 #define [FINISHED](#) (6)

2.1.1.8 #define [FOLLOW](#) (2)

2.1.1.9 #define [FOLLOW\\_LEFT](#) (1)

2.1.1.10 #define [FOLLOW\\_LINE](#) (1)

2.1.1.11 #define [FOLLOW\\_RIGHT](#) (0)

2.1.1.12 #define [FORWARD](#) (5)

2.1.1.13 #define [GREEN](#) (4)

2.1.1.14 #define [LEFT](#) (4)

2.1.1.15 #define [LIGHT\\_BLUE](#) (6)

2.1.1.16 #define [LIGHT\\_PORT](#) [NXT\\_PORT\\_S1](#)

2.1.1.17 #define [MOTOR\\_LEFT](#) [NXT\\_PORT\\_A](#)

[defines.h](#)

This file details all the defines used throughout the code.

Jack Mitchell <jgm11@le.ac.uk>

2.1.1.18 `#define MOTOR_RIGHT NXT_PORT_C`

2.1.1.19 `#define NORTH (0)`

2.1.1.20 `#define NUM_COLORS (7)`

2.1.1.21 `#define PAUSED (3)`

2.1.1.22 `#define RAND_MAX (3);`

2.1.1.23 `#define RED (3)`

2.1.1.24 `#define RIGHT (3)`

2.1.1.25 `#define SEARCH (1)`

2.1.1.26 `#define SEARCH_CORNER (4)`

2.1.1.27 `#define SOUTH (2)`

2.1.1.28 `#define START (0)`

2.1.1.29 `#define SWITCH_WAIT (3)`

2.1.1.30 `#define TEST (0)`

2.1.1.31 `#define TURN (5)`

2.1.1.32 `#define UNKNOWN (99)`

2.1.1.33 `#define UNSURE (0)`

2.1.1.34 `#define WEST (3)`

2.1.1.35 `#define WHITE (1)`

2.1.1.36 `#define YELLOW (2)`

## 2.2 functions.c File Reference

```
#include "kernel.h"
```

```
#include "kernel_id.h"
```

```
#include "ecrobot_interface.h"
```

```
#include "defines.h"
#include "functions.h"
```

## Functions

- void `set_both_motor_speed` (S8 left, S8 right)  
*Set both motor speeds in one easy function.*
- void `set_motor_threshold` (U8 l\_min, U8 l\_max, U8 r\_min, U8 r\_max)  
*Sets the upper and lower motor speed thresholds.*
- int `query_turn` (U8 x, U8 y)  
*Query if it is viable to make a turn according to Tremauxes' Algorithm.*
- int `change_position` (U8 check)  
*Change the position of the robot in the map array.*
- void `robot_follow` (U8 direction, U8 `line_side`)  
*sets the motor speeds according to the current colour detected*
- void `check_color` ()  
*queries the colour sensor and sets the global colour variables*

### 2.2.1 Function Documentation

#### 2.2.1.1 int `change_position` ( U8 *check* )

Change the position of the robot in the map array.

#### Parameters

<i>check</i>	if this is a dry run of an actual change of position
--------------	--

#### Returns

viable move according to Tremauxes' Algorithm

#### 2.2.1.2 void `check_color` ( )

queries the colour sensor and sets the global colour variables

#### 2.2.1.3 int `query_turn` ( U8 x, U8 y )

Query if it is viable to make a turn according to Tremauxes' Algorithm.

**Parameters**

<i>x</i>	proposed x co-ordinate
<i>y</i>	proposed y co-ordinate

**Returns**

viable move according to Tremauxes' Algorithm

**2.2.1.4 void robot\_follow ( U8 direction, U8 line\_side )**

sets the motor speeds according to the current colour detected

**Parameters**

<i>direction</i>	what mode the robot is in
<i>line_side</i>	the side of the line the robot should follow

**2.2.1.5 void set\_both\_motor\_speed ( S8 left, S8 right )**

Set both motor speeds in one easy function.

[functions.c](#)

Functions for control of the robot

Jack Mitchell <[jgm11@le.ac.uk](mailto:jgm11@le.ac.uk)>

**Parameters**

<i>left</i>	left motor speed
<i>right</i>	right motor speed

**2.2.1.6 void set\_motor\_threshold ( U8 l\_min, U8 l\_max, U8 r\_min, U8 r\_max )**

Sets the upper and lower motor speed thresholds.

**Parameters**

<i>l_min</i>	left motor lower
<i>l_max</i>	left motor upper
<i>r_min</i>	right motor lower
<i>r_max</i>	right motor upper

**2.3 functions.h File Reference**

```
#include "kernel.h"
```

```
#include "kernel_id.h"
#include "ecrobot_interface.h"
```

## Functions

- void [set\\_both\\_motor\\_speed](#) (S8 left, S8 right)  
*Set both motor speeds in one easy function.*
- void [set\\_motor\\_threshold](#) (U8 l\_min, U8 l\_max, U8 r\_min, U8 r\_max)  
*Sets the upper and lower motor speed thresholds.*
- int [query\\_turn](#) (U8 x, U8 y)  
*Query if it is viable to make a turn according to Tremauxes' Algorithm.*
- int [change\\_position](#) (U8 check)  
*Change the position of the robot in the map array.*
- void [robot\\_follow](#) (U8 direction, U8 [line\\_side](#))  
*sets the motor speeds according to the current colour detected*
- void [check\\_color](#) ()  
*queries the colour sensor and sets the global colour variables*

## Variables

- U8 [make\\_turn](#)
- U8 [checked](#) [6]
- U8 [old\\_color](#)
- U8 [color](#)
- S16 [colorCal](#) [8][6]
- S8 [motor\\_right](#)
- S8 [motor\\_left](#)
- U8 [map](#) [100][100]
- U8 [x\\_position](#) = 50
- U8 [y\\_position](#) = 50
- U8 [orientation](#) = NORTH

### 2.3.1 Function Documentation

#### 2.3.1.1 int [change\\_position](#) ( U8 *check* )

Change the position of the robot in the map array.

#### Parameters

<i>check</i>	if this is a dry run of an actual change of position
--------------	--

**Returns**

viable move according to Tremauxes' Algorithm

**2.3.1.2 void check\_color ( )**

queries the colour sensor and sets the global colour variables

**2.3.1.3 int query\_turn ( U8 x, U8 y )**

Query if it is viable to make a turn according to Tremauxes' Algorithm.

**Parameters**

<i>x</i>	proposed x co-ordinate
<i>y</i>	proposed y co-ordinate

**Returns**

viable move according to Tremauxes' Algorithm

**2.3.1.4 void robot\_follow ( U8 direction, U8 line\_side )**

sets the motor speeds according to the current colour detected

**Parameters**

<i>direction</i>	what mode the robot is in
<i>line_side</i>	the side of the line the robot should follow

**2.3.1.5 void set\_both\_motor\_speed ( S8 left, S8 right )**

Set both motor speeds in one easy function.

[functions.c](#)

Functions for control of the robot

Jack Mitchell <[jgm11@le.ac.uk](mailto:jgm11@le.ac.uk)>

**Parameters**

<i>left</i>	left motor speed
<i>right</i>	right motor speed

### 2.3.1.6 void set\_motor\_threshold ( U8 *l\_min*, U8 *l\_max*, U8 *r\_min*, U8 *r\_max* )

Sets the upper and lower motor speed thresholds.

#### Parameters

<i>l_min</i>	left motor lower
<i>l_max</i>	left motor upper
<i>r_min</i>	right motor lower
<i>r_max</i>	right motor upper

## 2.3.2 Variable Documentation

### 2.3.2.1 U8 checked[6]

### 2.3.2.2 U8 color

### 2.3.2.3 S16 colorCal[8][6]

### 2.3.2.4 U8 make\_turn

[functions.h](#)

Header file defining functions

Jack Mitchell <[jgm11@le.ac.uk](mailto:jgm11@le.ac.uk)>

### 2.3.2.5 U8 map[100][100]

### 2.3.2.6 S8 motor\_left

### 2.3.2.7 S8 motor\_right

### 2.3.2.8 U8 old\_color

### 2.3.2.9 U8 orientation = NORTH

### 2.3.2.10 U8 x\_position = 50

### 2.3.2.11 U8 y\_position = 50

## 2.4 kernel\_cfg.c File Reference

```
#include "osek_kernel.h"
#include "kernel_id.h"
#include "alarm.h"
#include "interrupt.h"
```



```
#include "resource.h"
#include "task.h"
```

## Defines

- #define `__STK_UNIT` VP
- #define `__TCOUNT_STK_UNIT`(sz) (((sz) + sizeof(\_\_STK\_UNIT) - 1) / sizeof(\_\_STK\_UNIT))
- #define `TNUM_ALARM` 3
- #define `TNUM_COUNTER` 1
- #define `TNUM_ISR2` 0
- #define `TNUM_RESOURCE` 0
- #define `TNUM_TASK` 3
- #define `TNUM_EXTTASK` 0
- #define `IPL_MAXISR2` 0

## Functions

- void TASKNAME() `ReadButtons` (void)
- void TASKNAME() `ReadColour` (void)
- void TASKNAME() `ManageState` (void)
- `DEFINE_CTXB` (TNUM\_TASK)
- `DeclareTask` (ReadColour)
- static void `_activate_alarm_cyclic_alarm_ReadColour` (void)
- `DeclareTask` (ManageState)
- static void `_activate_alarm_cyclic_alarm_ManageState` (void)
- `DeclareTask` (ReadButtons)
- static void `_activate_alarm_cyclic_alarm_ReadButtons` (void)
- void `object_initialize` (void)

## Variables

- const UINT8 `tnum_alarm` = TNUM\_ALARM
- const UINT8 `tnum_counter` = TNUM\_COUNTER
- const UINT8 `tnum_isr2` = TNUM\_ISR2
- const UINT8 `tnum_resource` = TNUM\_RESOURCE
- const UINT8 `tnum_task` = TNUM\_TASK
- const UINT8 `tnum_exttask` = TNUM\_EXTTASK
- const TaskType `ReadButtons` = 0
- const TaskType `ReadColour` = 1
- const TaskType `ManageState` = 2
- static `__STK_UNIT _stack_ReadButtons` [`__TCOUNT_STK_UNIT`(512)]
- static `__STK_UNIT _stack_ReadColour` [`__TCOUNT_STK_UNIT`(512)]
- static `__STK_UNIT _stack_ManageState` [`__TCOUNT_STK_UNIT`(512)]

- const Priority [tinib\\_inipri](#) [TNUM\_TASK] = { TPRI\_MINTASK + 4, TPRI\_MINTASK + 2, TPRI\_MINTASK + 5, }
- const Priority [tinib\\_exepr](#)i [TNUM\_TASK] = { TPRI\_MINTASK + 4, TPRI\_MINTASK + 2, TPRI\_MINTASK + 5, }
- const UINT8 [tinib\\_maxact](#) [TNUM\_TASK] = { (1) - 1, (1) - 1, (1) - 1, }
- const AppModeType [tinib\\_autoact](#) [TNUM\_TASK] = { 0x00000001, 0x00000000, 0x00000000, }
- const FP [tinib\\_task](#) [TNUM\_TASK] = { TASKNAME( ReadButtons ), TASKNAME( ReadColour ), TASKNAME( ManageState ), }
- const \_\_STK\_UNIT [tinib\\_stk](#) [TNUM\_TASK] = { (\_\_STK\_UNIT)\_stack\_ReadButtons, (\_\_STK\_UNIT)\_stack\_ReadColour, (\_\_STK\_UNIT)\_stack\_ManageState, }
- const UINT16 [tinib\\_stksz](#) [TNUM\_TASK] = { 512, 512, 512, }
- TaskType [tcb\\_next](#) [TNUM\_TASK]
- UINT8 [tcb\\_tstat](#) [TNUM\_TASK]
- Priority [tcb\\_curpri](#) [TNUM\_TASK]
- UINT8 [tcb\\_actent](#) [TNUM\_TASK]
- EventMaskType [tcb\\_curevt](#) [TNUM\_EXTTASK+1]
- EventMaskType [tcb\\_waievt](#) [TNUM\_EXTTASK+1]
- ResourceType [tcb\\_lastres](#) [TNUM\_TASK]
- const CounterType [SysTimerCnt](#) = 0
- const TickType [cntinib\\_maxval](#) [TNUM\_COUNTER] = { 10000, }
- const TickType [cntinib\\_maxval2](#) [TNUM\_COUNTER] = { 20001, }
- const TickType [cntinib\\_tickbase](#) [TNUM\_COUNTER] = { 1, }
- const TickType [cntinib\\_mincyc](#) [TNUM\_COUNTER] = { 1, }
- AlarmType [cntcb\\_almque](#) [TNUM\_COUNTER]
- TickType [cntcb\\_curval](#) [TNUM\_COUNTER]
- const AlarmType [cyclic\\_alarm\\_ReadColour](#) = 0
- const AlarmType [cyclic\\_alarm\\_ManageState](#) = 1
- const AlarmType [cyclic\\_alarm\\_ReadButtons](#) = 2
- const CounterType [alminib\\_cntid](#) [TNUM\_ALARM] = { 0, 0, 0, }
- const FP [alminib\\_cback](#) [TNUM\_ALARM] = { \_activate\_alarm\_cyclic\_alarm\_ReadColour, \_activate\_alarm\_cyclic\_alarm\_ManageState, \_activate\_alarm\_cyclic\_alarm\_ReadButtons, }
- const AppModeType [alminib\\_autosta](#) [TNUM\_ALARM] = { 0x00000001, 0x00000001, 0x00000001, }
- const TickType [alminib\\_almval](#) [TNUM\_ALARM] = { 1, 1, 1, }
- const TickType [alminib\\_cycle](#) [TNUM\_ALARM] = { 2, 10, 100, }
- AlarmType [almcb\\_next](#) [TNUM\_ALARM]
- AlarmType [almcb\\_prev](#) [TNUM\_ALARM]
- TickType [almcb\\_almval](#) [TNUM\_ALARM]
- TickType [almcb\\_cycle](#) [TNUM\_ALARM]
- const Priority [resinib\\_ceilpri](#) [TNUM\_RESOURCE+1] = { 0 }
- Priority [rescb\\_prevpri](#) [TNUM\_RESOURCE+1]
- ResourceType [rescb\\_prevres](#) [TNUM\_RESOURCE+1]
- const IPL [ipl\\_maxisr2](#) = IPL\_MAXISR2
- const Priority [isrinib\\_intpri](#) [TNUM\_ISR2+1] = { 0 }
- ResourceType [isrcb\\_lastres](#) [TNUM\_ISR2+1]



## 2.4.1 Define Documentation

2.4.1.1 `#define __STK_UNIT VP`

2.4.1.2 `#define __TCOUNT_STK_UNIT( sz ) (((sz) + sizeof(__STK_UNIT) - 1) / sizeof(__STK_UNIT))`

2.4.1.3 `#define IPL_MAXISR2 0`

2.4.1.4 `#define TNUM_ALARM 3`

2.4.1.5 `#define TNUM_COUNTER 1`

2.4.1.6 `#define TNUM_EXTTASK 0`

2.4.1.7 `#define TNUM_ISR2 0`

2.4.1.8 `#define TNUM_RESOURCE 0`

2.4.1.9 `#define TNUM_TASK 3`

## 2.4.2 Function Documentation

2.4.2.1 `static void _activate_alarm_cyclic_alarm_ManageState ( void ) [static]`

2.4.2.2 `static void _activate_alarm_cyclic_alarm_ReadButtons ( void ) [static]`

2.4.2.3 `static void _activate_alarm_cyclic_alarm_ReadColour ( void ) [static]`

2.4.2.4 `DeclareTask ( ReadButtons )`

2.4.2.5 `DeclareTask ( ReadColour )`

2.4.2.6 `DeclareTask ( ManageState )`

2.4.2.7 `DEFINE_CTXB ( TNUM_TASK )`

2.4.2.8 `void TASKNAME() ManageState ( void )`

2.4.2.9 `void object_initialize ( void )`

2.4.2.10 `void TASKNAME() ReadButtons ( void )`

2.4.2.11 `void TASKNAME() ReadColour ( void )`

## 2.4.3 Variable Documentation

2.4.3.1 `__STK_UNIT _stack_ManageState[__TCOUNT_STK_UNIT(512)] [static]`

2.4.3.2 `__STK_UNIT _stack_ReadButtons[__TCOUNT_STK_UNIT(512)] [static]`

2.4.3.3 `__STK_UNIT _stack_ReadColour[__TCOUNT_STK_UNIT(512)] [static]`

2.4.3.4 `TickType almcb_almval[TNUM_ALARM]`

2.4.3.5 `TickType almcb_cycle[TNUM_ALARM]`

2.4.3.6 `AlarmType almcb_next[TNUM_ALARM]`

2.4.3.7 `AlarmType almcb_prev[TNUM_ALARM]`

### 2.5.1 Define Documentation

2.5.1.1 `#define appmode1 (AppModeType)( 1U << 0 )`

## 2.6 lineFollower.c File Reference

Main code for an Autonomous Colour Line Following Robot with Area Mapping and Decision Making Abilities.

```
#include "kernel.h"
#include "kernel_id.h"
#include "ecrobot_interface.h"
#include "stdlib.h"
#include "defines.h"
#include "functions.h"
#include "lineFollower.h"
```

### Functions

- [DeclareCounter](#) (SysTimerCnt)
- [DeclareTask](#) (ReadColour)
- [DeclareTask](#) (ReadButtons)
- [DeclareTask](#) (ManageState)
- void [ecrobot\\_device\\_initialize](#) (void)
- void [ecrobot\\_device\\_terminate](#) (void)
- void [user\\_1ms\\_isr\\_type2](#) (void)
- [TASK](#) (ManageState)
- [TASK](#) (ReadButtons)
- [TASK](#) (ReadColour)

### 2.6.1 Detailed Description

Main code for an Autonomous Colour Line Following Robot with Area Mapping and Decision Making Abilities.

#### Author

Jack Mitchell

#### Version

#### Date

2011-02-21

## 2.6.2 Function Documentation

2.6.2.1 `DeclareCounter ( SysTimerCnt )`

2.6.2.2 `DeclareTask ( ReadColour )`

2.6.2.3 `DeclareTask ( ManageState )`

2.6.2.4 `DeclareTask ( ReadButtons )`

2.6.2.5 `void ecrobot_device_initialize ( void )`

2.6.2.6 `void ecrobot_device_terminate ( void )`

2.6.2.7 `TASK ( ReadButtons )`

2.6.2.8 `TASK ( ReadColour )`

2.6.2.9 `TASK ( ManageState )`

2.6.2.10 `void user_1ms_isr_type2 ( void )`

## 2.7 lineFollower.h File Reference

### Variables

- U8 `state` = 0
- U8 `old_color`
- U8 `color_cal_state` = 0
- U8 `color`
- U32 `enter_button` = 0
- U32 `run_button` = 0
- S16 `colorCal` [8][6]
- U8 `random` = 0
- U8 `line_side` = FOLLOW\_LEFT
- U8 `make_turn`
- U8 `line_found`
- U16 `search_time` = 0
- U16 `black_count` = 0
- U8 `checked` [6]

### 2.7.1 Variable Documentation

2.7.1.1 U16 `black_count` = 0

2.7.1.2 U8 `checked`[6]

2.7.1.3 U8 `color`

2.7.1.4 U8 `color_cal_state` = 0

2.7.1.5 S16 `colorCal`[8][6]

2.7.1.6 U32 `enter_button` = 0

2.7.1.7 U8 `line_found`

2.7.1.8 U8 `line_side` = FOLLOW\_LEFT

2.7.1.9 U8 `make_turn`

[functions.h](#)

Header file defining functions

Jack Mitchell <[jgm11@le.ac.uk](mailto:jgm11@le.ac.uk)>

2.7.1.10 U8 `old_color`

2.7.1.11 U8 `random` = 0

2.7.1.12 U32 `run_button` = 0

2.7.1.13 U16 `search_time` = 0

2.7.1.14 U8 `state` = 0

[lineFollower.h](#)

Holds static variables

Jack Mitchell <[jgm11@le.ac.uk](mailto:jgm11@le.ac.uk)>

# Index

\_\_STK\_UNIT  
    kernel\_cfg.c, 14  
\_\_TCOUNT\_STK\_UNIT  
    kernel\_cfg.c, 14  
\_activate\_alarm\_cyclic\_alarm\_ManageState  
    kernel\_cfg.c, 14  
\_activate\_alarm\_cyclic\_alarm\_ReadButtons  
    kernel\_cfg.c, 14  
\_activate\_alarm\_cyclic\_alarm\_ReadColour  
    kernel\_cfg.c, 14  
\_stack\_ManageState  
    kernel\_cfg.c, 14  
\_stack\_ReadButtons  
    kernel\_cfg.c, 14  
\_stack\_ReadColour  
    kernel\_cfg.c, 14  
  
almcb\_almval  
    kernel\_cfg.c, 14  
almcb\_cycle  
    kernel\_cfg.c, 14  
almcb\_next  
    kernel\_cfg.c, 14  
almcb\_prev  
    kernel\_cfg.c, 14  
alminib\_almval  
    kernel\_cfg.c, 14  
alminib\_autosta  
    kernel\_cfg.c, 14  
alminib\_cback  
    kernel\_cfg.c, 14  
alminib\_cntid  
    kernel\_cfg.c, 14  
alminib\_cycle  
    kernel\_cfg.c, 14  
ALTER  
    defines.h, 4  
appmodel  
    kernel\_id.h, 15  
  
BLACK  
    defines.h, 4  
black\_count  
    lineFollower.h, 17  
BLUE  
    defines.h, 4  
CALIBRATE  
    defines.h, 4  
change\_position  
    functions.c, 6  
    functions.h, 8  
check\_color  
    functions.c, 6  
    functions.h, 9  
checked  
    functions.h, 10  
    lineFollower.h, 17  
cntcb\_almque  
    kernel\_cfg.c, 14  
cntcb\_curval  
    kernel\_cfg.c, 14  
cntinib\_maxval  
    kernel\_cfg.c, 14  
cntinib\_maxval2  
    kernel\_cfg.c, 14  
cntinib\_mincyc  
    kernel\_cfg.c, 14  
cntinib\_tickbase  
    kernel\_cfg.c, 14  
color  
    functions.h, 10  
    lineFollower.h, 17  
color\_cal\_state  
    lineFollower.h, 17  
colorCal  
    functions.h, 10  
    lineFollower.h, 17  
CORNER  
    defines.h, 4  
cyclic\_alarm\_ManageState  
    kernel\_cfg.c, 14



- cyclic\_alarm\_ReadButtons
  - kernel\_cfg.c, [14](#)
- cyclic\_alarm\_ReadColour
  - kernel\_cfg.c, [14](#)
- DeclareCounter
  - lineFollower.c, [16](#)
- DeclareTask
  - kernel\_cfg.c, [14](#)
  - lineFollower.c, [16](#)
- DEFINE\_CTXB
  - kernel\_cfg.c, [14](#)
- defines.h, [3](#)
  - ALTER, [4](#)
  - BLACK, [4](#)
  - BLUE, [4](#)
  - CALIBRATE, [4](#)
  - CORNER, [4](#)
  - EAST, [4](#)
  - FINISHED, [4](#)
  - FOLLOW, [4](#)
  - FOLLOW\_LEFT, [4](#)
  - FOLLOW\_LINE, [4](#)
  - FOLLOW\_RIGHT, [4](#)
  - FORWARD, [4](#)
  - GREEN, [4](#)
  - LEFT, [4](#)
  - LIGHT\_BLUE, [4](#)
  - LIGHT\_PORT, [4](#)
  - MOTOR\_LEFT, [4](#)
  - MOTOR\_RIGHT, [5](#)
  - NORTH, [5](#)
  - NUM\_COLORS, [5](#)
  - PAUSED, [5](#)
  - RAND\_MAX, [5](#)
  - RED, [5](#)
  - RIGHT, [5](#)
  - SEARCH, [5](#)
  - SEARCH\_CORNER, [5](#)
  - SOUTH, [5](#)
  - START, [5](#)
  - SWITCH\_WAIT, [5](#)
  - TEST, [5](#)
  - TURN, [5](#)
  - UNKNOWN, [5](#)
  - UNSURE, [5](#)
  - WEST, [5](#)
  - WHITE, [5](#)
  - YELLOW, [5](#)
- EAST
  - defines.h, [4](#)
- ecrobot\_device\_initialize
  - lineFollower.c, [16](#)
- ecrobot\_device\_terminate
  - lineFollower.c, [16](#)
- enter\_button
  - lineFollower.h, [17](#)
- FINISHED
  - defines.h, [4](#)
- FOLLOW
  - defines.h, [4](#)
- FOLLOW\_LEFT
  - defines.h, [4](#)
- FOLLOW\_LINE
  - defines.h, [4](#)
- FOLLOW\_RIGHT
  - defines.h, [4](#)
- FORWARD
  - defines.h, [4](#)
- functions.c, [5](#)
  - change\_position, [6](#)
  - check\_color, [6](#)
  - query\_turn, [6](#)
  - robot\_follow, [7](#)
  - set\_both\_motor\_speed, [7](#)
  - set\_motor\_threshold, [7](#)
- functions.h, [7](#)
  - change\_position, [8](#)
  - check\_color, [9](#)
  - checked, [10](#)
  - color, [10](#)
  - colorCal, [10](#)
  - make\_turn, [10](#)
  - map, [10](#)
  - motor\_left, [10](#)
  - motor\_right, [10](#)
  - old\_color, [10](#)
  - orientation, [10](#)
  - query\_turn, [9](#)
  - robot\_follow, [9](#)
  - set\_both\_motor\_speed, [9](#)
  - set\_motor\_threshold, [9](#)
  - x\_position, [10](#)
  - y\_position, [10](#)
- GREEN
  - defines.h, [4](#)
- IPL\_MAXISR2

- kernel\_cfg.c, [14](#)
- ipl\_maxisr2
  - kernel\_cfg.c, [14](#)
- isrcb\_lastres
  - kernel\_cfg.c, [14](#)
- isrinib\_intpri
  - kernel\_cfg.c, [14](#)
- kernel\_cfg.c, [10](#)
  - \_\_STK\_UNIT, [14](#)
  - \_\_TCOUNT\_STK\_UNIT, [14](#)
  - \_activate\_alarm\_cyclic\_alarm\_ManageState, [14](#)
  - \_activate\_alarm\_cyclic\_alarm\_ReadButtons, [14](#)
  - \_activate\_alarm\_cyclic\_alarm\_ReadColour, [14](#)
  - \_stack\_ManageState, [14](#)
  - \_stack\_ReadButtons, [14](#)
  - \_stack\_ReadColour, [14](#)
  - almcb\_almval, [14](#)
  - almcb\_cycle, [14](#)
  - almcb\_next, [14](#)
  - almcb\_prev, [14](#)
  - alminib\_almval, [14](#)
  - alminib\_autosta, [14](#)
  - alminib\_cback, [14](#)
  - alminib\_cntid, [14](#)
  - alminib\_cycle, [14](#)
  - cntcb\_almque, [14](#)
  - cntcb\_curval, [14](#)
  - cntinib\_maxval, [14](#)
  - cntinib\_maxval2, [14](#)
  - cntinib\_mincyc, [14](#)
  - cntinib\_tickbase, [14](#)
  - cyclic\_alarm\_ManageState, [14](#)
  - cyclic\_alarm\_ReadButtons, [14](#)
  - cyclic\_alarm\_ReadColour, [14](#)
  - DeclareTask, [14](#)
  - DEFINE\_CTXB, [14](#)
  - IPL\_MAXISR2, [14](#)
  - ipl\_maxisr2, [14](#)
  - isrcb\_lastres, [14](#)
  - isrinib\_intpri, [14](#)
  - ManageState, [14](#)
  - object\_initialize, [14](#)
  - ReadButtons, [14](#)
  - ReadColour, [14](#)
  - rescb\_prevpri, [14](#)
  - rescb\_prevres, [14](#)
  - resinib\_ceilpri, [14](#)
  - SysTimerCnt, [14](#)
  - tcb\_actcnt, [14](#)
  - tcb\_curevt, [14](#)
  - tcb\_curpri, [14](#)
  - tcb\_lastres, [14](#)
  - tcb\_next, [14](#)
  - tcb\_tstat, [14](#)
  - tcb\_waievt, [14](#)
  - tinib\_autoact, [14](#)
  - tinib\_exepr, [14](#)
  - tinib\_inipri, [14](#)
  - tinib\_maxact, [14](#)
  - tinib\_stk, [14](#)
  - tinib\_stksz, [14](#)
  - tinib\_task, [14](#)
  - TNUM\_ALARM, [14](#)
  - tnum\_alarm, [14](#)
  - TNUM\_COUNTER, [14](#)
  - tnum\_counter, [14](#)
  - TNUM\_EXTTASK, [14](#)
  - tnum\_exttask, [14](#)
  - TNUM\_ISR2, [14](#)
  - tnum\_isr2, [14](#)
  - TNUM\_RESOURCE, [14](#)
  - tnum\_resource, [14](#)
  - TNUM\_TASK, [14](#)
  - tnum\_task, [14](#)
- kernel\_id.h, [14](#)
  - appmode1, [15](#)
- LEFT
  - defines.h, [4](#)
- LIGHT\_BLUE
  - defines.h, [4](#)
- LIGHT\_PORT
  - defines.h, [4](#)
- line\_found
  - lineFollower.h, [17](#)
- line\_side
  - lineFollower.h, [17](#)
- lineFollower.c, [15](#)
  - DeclareCounter, [16](#)
  - DeclareTask, [16](#)
  - ecrobot\_device\_initialize, [16](#)
  - ecrobot\_device\_terminate, [16](#)
  - TASK, [16](#)
  - user\_1ms\_isr\_type2, [16](#)
- lineFollower.h, [16](#)
  - black\_count, [17](#)

- checked, [17](#)
- color, [17](#)
- color\_cal\_state, [17](#)
- colorCal, [17](#)
- enter\_button, [17](#)
- line\_found, [17](#)
- line\_side, [17](#)
- make\_turn, [17](#)
- old\_color, [17](#)
- random, [17](#)
- run\_button, [17](#)
- search\_time, [17](#)
- state, [17](#)
- make\_turn
  - functions.h, [10](#)
  - lineFollower.h, [17](#)
- ManageState
  - kernel\_cfg.c, [14](#)
- map
  - functions.h, [10](#)
- MOTOR\_LEFT
  - defines.h, [4](#)
- motor\_left
  - functions.h, [10](#)
- MOTOR\_RIGHT
  - defines.h, [5](#)
- motor\_right
  - functions.h, [10](#)
- NORTH
  - defines.h, [5](#)
- NUM\_COLORS
  - defines.h, [5](#)
- object\_initialize
  - kernel\_cfg.c, [14](#)
- old\_color
  - functions.h, [10](#)
  - lineFollower.h, [17](#)
- orientation
  - functions.h, [10](#)
- PAUSED
  - defines.h, [5](#)
- query\_turn
  - functions.c, [6](#)
  - functions.h, [9](#)
- RAND\_MAX
  - defines.h, [5](#)
- random
  - lineFollower.h, [17](#)
- ReadButtons
  - kernel\_cfg.c, [14](#)
- ReadColour
  - kernel\_cfg.c, [14](#)
- RED
  - defines.h, [5](#)
- rescb\_prevpri
  - kernel\_cfg.c, [14](#)
- rescb\_prevres
  - kernel\_cfg.c, [14](#)
- resinib\_ceilpri
  - kernel\_cfg.c, [14](#)
- RIGHT
  - defines.h, [5](#)
- robot\_follow
  - functions.c, [7](#)
  - functions.h, [9](#)
- run\_button
  - lineFollower.h, [17](#)
- SEARCH
  - defines.h, [5](#)
- SEARCH\_CORNER
  - defines.h, [5](#)
- search\_time
  - lineFollower.h, [17](#)
- set\_both\_motor\_speed
  - functions.c, [7](#)
  - functions.h, [9](#)
- set\_motor\_threshold
  - functions.c, [7](#)
  - functions.h, [9](#)
- SOUTH
  - defines.h, [5](#)
- START
  - defines.h, [5](#)
- state
  - lineFollower.h, [17](#)
- SWITCH\_WAIT
  - defines.h, [5](#)
- SysTimerCnt
  - kernel\_cfg.c, [14](#)
- TASK
  - lineFollower.c, [16](#)
- tcb\_actcnt
  - kernel\_cfg.c, [14](#)

- tcb\_curevt
  - kernel\_cfg.c, [14](#)
- tcb\_curpri
  - kernel\_cfg.c, [14](#)
- tcb\_lastres
  - kernel\_cfg.c, [14](#)
- tcb\_next
  - kernel\_cfg.c, [14](#)
- tcb\_tstat
  - kernel\_cfg.c, [14](#)
- tcb\_waievt
  - kernel\_cfg.c, [14](#)
- TEST
  - defines.h, [5](#)
- tinib\_autoact
  - kernel\_cfg.c, [14](#)
- tinib\_exeprpri
  - kernel\_cfg.c, [14](#)
- tinib\_inipri
  - kernel\_cfg.c, [14](#)
- tinib\_maxact
  - kernel\_cfg.c, [14](#)
- tinib\_stk
  - kernel\_cfg.c, [14](#)
- tinib\_stksz
  - kernel\_cfg.c, [14](#)
- tinib\_task
  - kernel\_cfg.c, [14](#)
- TNUM\_ALARM
  - kernel\_cfg.c, [14](#)
- tnum\_alarm
  - kernel\_cfg.c, [14](#)
- TNUM\_COUNTER
  - kernel\_cfg.c, [14](#)
- tnum\_counter
  - kernel\_cfg.c, [14](#)
- TNUM\_EXTTASK
  - kernel\_cfg.c, [14](#)
- tnum\_exttask
  - kernel\_cfg.c, [14](#)
- TNUM\_ISR2
  - kernel\_cfg.c, [14](#)
- tnum\_isr2
  - kernel\_cfg.c, [14](#)
- TNUM\_RESOURCE
  - kernel\_cfg.c, [14](#)
- tnum\_resource
  - kernel\_cfg.c, [14](#)
- TNUM\_TASK
  - kernel\_cfg.c, [14](#)
- tnum\_task
  - kernel\_cfg.c, [14](#)
- TURN
  - defines.h, [5](#)
- UNKNOWN
  - defines.h, [5](#)
- UNSURE
  - defines.h, [5](#)
- user\_1ms\_isr\_type2
  - lineFollower.c, [16](#)
- WEST
  - defines.h, [5](#)
- WHITE
  - defines.h, [5](#)
- x\_position
  - functions.h, [10](#)
- y\_position
  - functions.h, [10](#)
- YELLOW
  - defines.h, [5](#)