

Alphabot

Generated by Doxygen 1.8.13

## Contents

<b>1</b>	<b>C++ Alphabot API for the Raspberry PI</b>	<b>1</b>
<b>2</b>	<b>Hierarchical Index</b>	<b>4</b>
2.1	Class Hierarchy . . . . .	4
<b>3</b>	<b>Class Index</b>	<b>4</b>
3.1	Class List . . . . .	4
<b>4</b>	<b>Class Documentation</b>	<b>4</b>
4.1	AlphaBot Class Reference . . . . .	4
4.1.1	Detailed Description . . . . .	5
4.1.2	Constructor & Destructor Documentation . . . . .	5
4.1.3	Member Function Documentation . . . . .	5
4.1.4	Member Data Documentation . . . . .	8
4.2	AlphaBot::StepCallback Class Reference . . . . .	9
4.2.1	Detailed Description . . . . .	9
4.2.2	Member Function Documentation . . . . .	9
	<b>Index</b>	<b>11</b>

## 1 C++ Alphabot API for the Raspberry PI

C++ class to control the basic functionality of the Alphabot (<https://www.waveshare.com/wiki/AlphaBot>) with a Raspberry PI:

- PWM motor control
- Distance sensor readings (digital and analogue)
- Battery voltage
- Wheel speed encoders (rotating flags)
- IR sensor readings
- 10Hz sampling rate with callback

## Prerequisites

### pigpio

The Raspberry PI hardware is accessed via the `pigpio` library. Install its package:

```
apt-get install libpigpio-dev
```

### Timer library

Install the timer library from: <https://github.com/berndporr/cppTimer>

### ncurses

The demo programs below display the sensor readings with the `ncurses` library. Install it with

```
apt-get install libncurses-dev
```

## Building

The built system is `cmake`. Just type:

```
cmake .  
make
```

## Usage

### Start/stop

Start the communication with the robot:

```
start()
```

Stop the communication:

```
stop()
```

### Motor speed

Setting the speeds of the left/right wheels:

```
setLeftWheelSpeed(float speed);  
setRightWheelSpeed(float speed);
```

where `speed` is between -1 and +1.

### Get Collision sensor readings

The digital functions are boolean values which are true if the distance sensor threshold (set with the potentiometer) has been reached:

```
bool getCollisionLeft();
bool getCollisionRight();
```

The analogue readings from the sensors from 0..1 can be read with:

```
float getLeftDistance()
float getRightDistance()
```

### Battery voltage

The function `getBatteryLevel()` provides the voltage of the battery in volt.

### Speed encoders

These are terrible encoders and really only reliable to detect if the wheel is spinning at all or not. Updates every 1/2 sec.

```
bool getLeftWheelSpinning();
bool getRightWheelSpinning();
```

### Infrared channels / general purpose ADC channels 0-4

```
float (&getIR())[nIR]
```

returns a reference to the whole array of IR sensor readings normalised between 0..1.

### Callback

Whenever a new set of analogue readings is available the callback `step` in `StepCallback` is called. It contains the reference to the Alphabet class itself so that one can read the different ADC values.

### Demo programs

`testIO` is a simple test program which displays the different sensor readings and you can test the motors. It also shows how the callback is used to display the sensor readings.

`testMotor` ramps up the motor speed and back again.

`obstacleAvoid.cpp` implements a Braitenberg vehicle and it has an emergency stop if the digital readings from the collision sensors are triggered. This is an instructional example and needs to be tweaked. The fun begins here! :)

## 2 Hierarchical Index

### 2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

CppTimer

**AlphaBot** 4

**AlphaBot::StepCallback** 9

## 3 Class Index

### 3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

[AlphaBot](#) 4

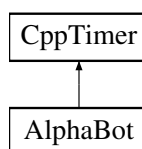
[AlphaBot::StepCallback](#) 9

## 4 Class Documentation

### 4.1 AlphaBot Class Reference

```
#include <alphabot.h>
```

Inheritance diagram for AlphaBot:



#### Classes

- class [StepCallback](#)

### Public Member Functions

- void `start` (long `_samplingInterval`=100)
- void `stop` ()
- `~AlphaBot` ()
- void `registerStepCallback` (`StepCallback` \*`_stepcallback`)
- void `setLeftWheelSpeed` (float `speed`)
- void `setRightWheelSpeed` (float `speed`)
- bool `getLeftWheelSpinning` ()
- bool `getRightWheelSpinning` ()
- float `getBatteryLevel` ()
- bool `getCollisionLeft` ()
- bool `getCollisionRight` ()
- float `getLeftDistance` ()
- float `getRightDistance` ()
- const float(& `getIR` ()) [`nIR`]

### Static Public Attributes

- static const unsigned `nIR` = 5

#### 4.1.1 Detailed Description

Alphabot class which communicates with the Alphabot hardware

#### 4.1.2 Constructor & Destructor Documentation

##### 4.1.2.1 `~AlphaBot()`

```
AlphaBot::~AlphaBot ( ) [inline]
```

Destroys the Alpha Bot object and stops any communication

#### 4.1.3 Member Function Documentation

##### 4.1.3.1 `getBatteryLevel()`

```
float AlphaBot::getBatteryLevel ( ) [inline]
```

Get the Battery Level

### Returns

float Battery level in Volt

#### 4.1.3.2 `getCollisionLeft()`

```
bool AlphaBot::getCollisionLeft ( )
```

Get the Collision Left (digital signal from the distance sensor)

##### Returns

true Obstacle detected  
false No Obstacle

#### 4.1.3.3 `getCollisionRight()`

```
bool AlphaBot::getCollisionRight ( )
```

Get the Collision Right (digital signal from the distance sensor)

##### Returns

true Obstacle detected  
false No Obstacle

#### 4.1.3.4 `getIR()`

```
const float (& AlphaBot::getIR ( ))[nIR] [inline]
```

Returns the IR channels as a reference to an array with length nIR. This allows using C+11 style loops.

##### Returns

Reference to a const array with the 5 IR readings (0..1)

#### 4.1.3.5 `getLeftDistance()`

```
float AlphaBot::getLeftDistance ( ) [inline]
```

Get the Left Distance

##### Returns

float Value between 0 and 1 as relative distance.

#### 4.1.3.6 getLeftWheelSpinning()

```
bool AlphaBot::getLeftWheelSpinning ( ) [inline]
```

Info if the left wheel is spinning.

##### Returns

true Wheel is spinning  
false While is not spinning

#### 4.1.3.7 getRightDistance()

```
float AlphaBot::getRightDistance ( ) [inline]
```

Get the Right Distance

##### Returns

float Value between 0 and 1 as relative distance.

#### 4.1.3.8 getRightWheelSpinning()

```
bool AlphaBot::getRightWheelSpinning ( ) [inline]
```

Info if the right wheel is spinning.

##### Returns

true Wheel is spinning  
false While is not spinning

#### 4.1.3.9 registerStepCallback()

```
void AlphaBot::registerStepCallback (
    StepCallback * _stepcallback ) [inline]
```

registers callback which signals new data

##### Parameters

<code>_stepcallback</code>	A pointer to the callback interface
----------------------------	-------------------------------------



#### 4.1.3.10 setLeftWheelSpeed()

```
void AlphaBot::setLeftWheelSpeed (
    float speed )
```

Set the Left Wheel Speed

##### Parameters

<i>speed</i>	between -1 and +1
--------------	-------------------

#### 4.1.3.11 setRightWheelSpeed()

```
void AlphaBot::setRightWheelSpeed (
    float speed )
```

Set the Right Wheel Speed

##### Parameters

<i>speed</i>	between -1 and +1
--------------	-------------------

#### 4.1.3.12 start()

```
void AlphaBot::start (
    long _samplingInterval = 100 )
```

Starts the communication with the robot.

##### Parameters

<i>_samplingInterval</i>	Sampling interval for the ADC data
--------------------------	------------------------------------

#### 4.1.3.13 stop()

```
void AlphaBot::stop ( )
```

Stops the communication with the Alphabot

#### 4.1.4 Member Data Documentation

#### 4.1.4.1 nIR

```
const unsigned AlphaBot::nIR = 5 [static]
```

Number of IR channels or anagloue input channels at the rear of the robot.

The documentation for this class was generated from the following file:

- alphabot.h

## 4.2 AlphaBot::StepCallback Class Reference

```
#include <alphabot.h>
```

### Public Member Functions

- virtual void [step](#) (AlphaBot &)=0

#### 4.2.1 Detailed Description

Callback interface which is called at the specified sampling rate. The method step needs to be implemented.

#### 4.2.2 Member Function Documentation

##### 4.2.2.1 step()

```
virtual void AlphaBot::StepCallback::step (  
    AlphaBot & ) [pure virtual]
```

Called at the specified sampling rate with a reference to the Alphabot instance which has all the getters for the data and setters for the motor control.

The documentation for this class was generated from the following file:

- alphabot.h



## Index

~AlphaBot

AlphaBot, [5](#)

AlphaBot, [4](#)

~AlphaBot, [5](#)

getBatteryLevel, [5](#)

getCollisionLeft, [5](#)

getCollisionRight, [6](#)

getIR, [6](#)

getLeftDistance, [6](#)

getLeftWheelSpinning, [6](#)

getRightDistance, [7](#)

getRightWheelSpinning, [7](#)

nIR, [8](#)

registerStepCallback, [7](#)

setLeftWheelSpeed, [7](#)

setRightWheelSpeed, [8](#)

start, [8](#)

stop, [8](#)

AlphaBot::StepCallback, [9](#)

step, [9](#)

getBatteryLevel

AlphaBot, [5](#)

getCollisionLeft

AlphaBot, [5](#)

getCollisionRight

AlphaBot, [6](#)

getIR

AlphaBot, [6](#)

getLeftDistance

AlphaBot, [6](#)

getLeftWheelSpinning

AlphaBot, [6](#)

getRightDistance

AlphaBot, [7](#)

getRightWheelSpinning

AlphaBot, [7](#)

nIR

AlphaBot, [8](#)

registerStepCallback

AlphaBot, [7](#)

setLeftWheelSpeed

AlphaBot, [7](#)

setRightWheelSpeed

AlphaBot, [8](#)

start

AlphaBot, [8](#)

step

AlphaBot::StepCallback, [9](#)

stop

AlphaBot, [8](#)