FRC 2018 Software Documentation

Team 5572: The ROSBOTS

Contents

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Namespace Documentation

4.1 field Namespace Reference

Namespaces

- side

4.2 field::side Namespace Reference

Functions

- void setup ()
- bool switch_near ()
- bool switch_far ()
- bool scale ()

Variables

- const bool left = false
- const bool right = true

4.2.1 Function Documentation

4.2.1.1 bool field::side::scale ( ) [inline]

4.2.1.2 void field::side::setup ( ) [inline]

4.2.1.3 bool field::side::switch_far ( ) [inline]

4.2.1.4 bool field::side::switch_near ( ) [inline]

4.2.2 Variable Documentation

4.2.2.1 const bool field::side::left = false

4.2.2.2 const bool field::side::right = true

Chapter 5

Class Documentation

5.1 Curve Struct Reference

Describes the position and direction of a robot after a curve amount.

```
#include <drivetrain.h>
```

Public Attributes

- double x

    Horizontal Position.
- double y

    Vertical Position.
- double heading

    Direction of the robot in radians.

5.1.1 Detailed Description

Describes the position and direction of a robot after a curve amount.

5.1.2 Member Data Documentation

5.1.2.1 double Curve::heading

Direction of the robot in radians.

5.1.2.2 double Curve::x

Horizontal Position.

5.1.2.3 double Curve::y

Vertical Position.

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

- src/drivetrain.h

5.2 DoublePair Struct Reference

Stores generic 2-value real number objects.

```
#include <drivetrain.h>
```

Public Attributes

- double u

  First Value.
- double v

  Second Value.

5.2.1 Detailed Description

Stores generic 2-value real number objects.

Examples of usage are 2d coordinates and differential drive outputs.

5.2.2 Member Data Documentation

5.2.2.1 double DoublePair::u

First Value.

5.2.2.2 double DoublePair::v

Second Value.

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

- src/drivetrain.h

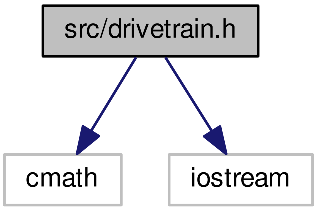
Chapter 6

File Documentation

6.1 src/drivetrain.h File Reference

```
#include <cmath>
#include <iostream>
```

Include dependency graph for drivetrain.h:



Classes

- struct Curve

  Describes the position and direction of a robot after a curve amount.
- struct DoublePair

  Stores generic 2-value real number objects.

Macros

- #define HALF_PI 1.570796327

Functions

- Curve operator+ (Curve a, Curve b)

  Adds two curves.
- Curve curveamount (double L1, double L2, double W)

  Turn two encoder readings (in coordinate units) into coordinates.

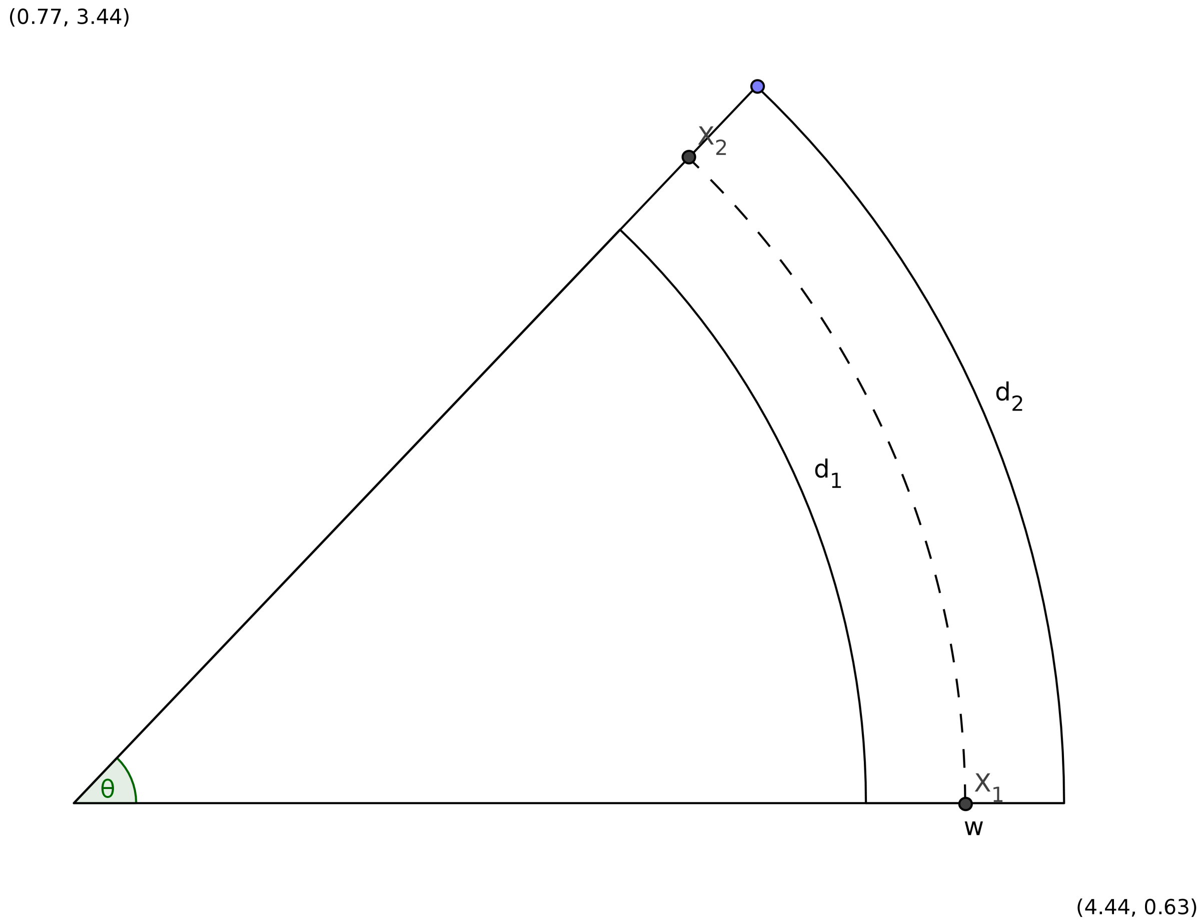
6.1.1 Macro Definition Documentation

6.1.1.1 #define HALF_PI 1.570796327

6.1.2 Function Documentation

6.1.2.1 Curve curveamount (double *L1*, double *L2*, double *W*) [inline]

Turn two encoder readings (in coordinate units) into coordinates.



$$r = \frac{y}{sin(\theta)} \tag{6.1}$$

$$x = \frac{y}{sin(\theta)} \tag{6.2}$$

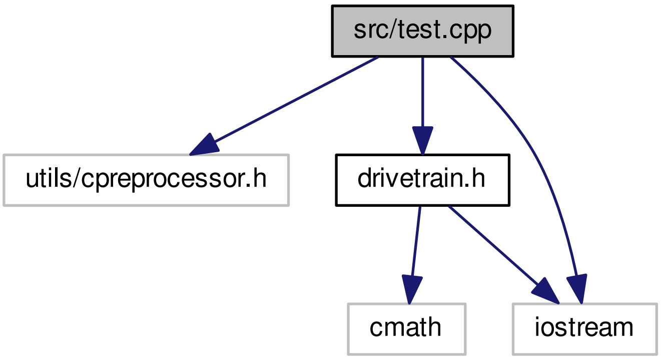
6.1.2.2 Curve operator+ (Curve *a*, Curve *b*) [inline]

Adds two curves.

6.2 src/test.cpp File Reference

```
#include "utils/cpreprocessor.h"
#include "utils/kernel_interface.h"
#include <iostream>
```

Include dependency graph for test.cpp:



Macros

- #define ROOT_REQUIRED
- #define ROOT_PASS "socrates"

Functions

- int main ()

6.2.1 Macro Definition Documentation

6.2.1.1 #define ROOT_PASS "socrates"

6.2.1.2 #define ROOT_REQUIRED

6.2.2 Function Documentation

6.2.2.1 int main ( )

6.3 src/utils/kernel_interface.h File Reference

This graph shows which files directly or indirectly include this file:

6.4 src/utils/linux.cpp File Reference

```
#include "kernel_interface.h"
#include <unistd.h>
#include <iostream>
#include <string.h>
#include <pwd.h>
#include <sys/wait.h>
#include <stdio.h>
#include <signal.h>
```

Include dependency graph for linux.cpp:

Macros

- #define ROOT_NAME "root"
- #define WRITE_END 1
- #define READ_END 0

Functions

- void signal_register (int a, void(∗v)(int))

  Creates a signal handler.
- void root_mode (char ∗argv[ ], const char ∗password)

  Resets program to be in sudo mode.

6.4.1 Macro Definition Documentation

6.4.1.1 #define READ_END 0

6.4.1.2 #define ROOT_NAME "root"

6.4.1.3 #define WRITE_END 1

6.4.2 Function Documentation

6.4.2.1 void root_mode (char ∗ *argv[ ],* const char ∗ *password*)

Resets program to be in sudo mode.

If the program is already root, it will instead just return

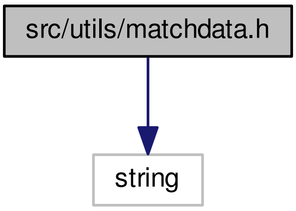
6.4.2.2 void signal_register (int *a,* void(∗)(int) *v*)

Creates a signal handler.

6.5 src/utils/matchdata.h File Reference

```
#include <string>
```

Include dependency graph for matchdata.h:



Namespaces

- field
- field::side

Functions

- void field::side::setup ()
- bool field::side::switch_near ()
- bool field::side::switch_far ()
- bool field::side::scale ()

Variables

- const bool field::side::left = false
- const bool field::side::right = true

Index