

FRC 2018 Software Documentation

Team 5572: The ROSBOTS

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	Class Index	3
2.1	Class List	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	field Namespace Reference	7
4.2	field::side Namespace Reference	7
4.2.1	Function Documentation	7
4.2.1.1	scale()	7
4.2.1.2	setup()	7
4.2.1.3	switch_far()	7
4.2.1.4	switch_near()	7
4.2.2	Variable Documentation	7
4.2.2.1	left	7
4.2.2.2	right	7

5	Class Documentation	9
5.1	Curve Struct Reference	9
5.1.1	Detailed Description	9
5.1.2	Member Data Documentation	9
5.1.2.1	heading	9
5.1.2.2	x	9
5.1.2.3	y	10
5.2	DoublePair Struct Reference	10
5.2.1	Detailed Description	10
5.2.2	Member Data Documentation	10
5.2.2.1	u	10
5.2.2.2	v	10
6	File Documentation	11
6.1	src/drivetrain.h File Reference	11
6.1.1	Macro Definition Documentation	12
6.1.1.1	HALF_PI	12
6.1.2	Function Documentation	12
6.1.2.1	curveamount(double L1, double L2, double W)	12
6.1.2.2	operator+(Curve a, Curve b)	13
6.2	src/test.cpp File Reference	13
6.2.1	Function Documentation	13
6.2.1.1	main()	13
6.3	src/utils/matchdata.h File Reference	14
	Index	15

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

field	7
field::side	7

Chapter 2

Class Index

2.1 Class List

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

Curve	Describes the position and direction of a robot after a curve amount	9
DoublePair	Stores generic 2-value real number objects	10

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

src/ drivetrain.h	11
src/ test.cpp	13
src/utls/ matchdata.h	14

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](#)

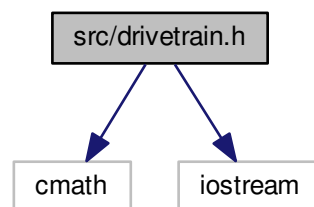
Chapter 6

File Documentation

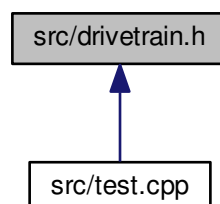
6.1 src/drivetrain.h File Reference

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

Include dependency graph for drivetrain.h:



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



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

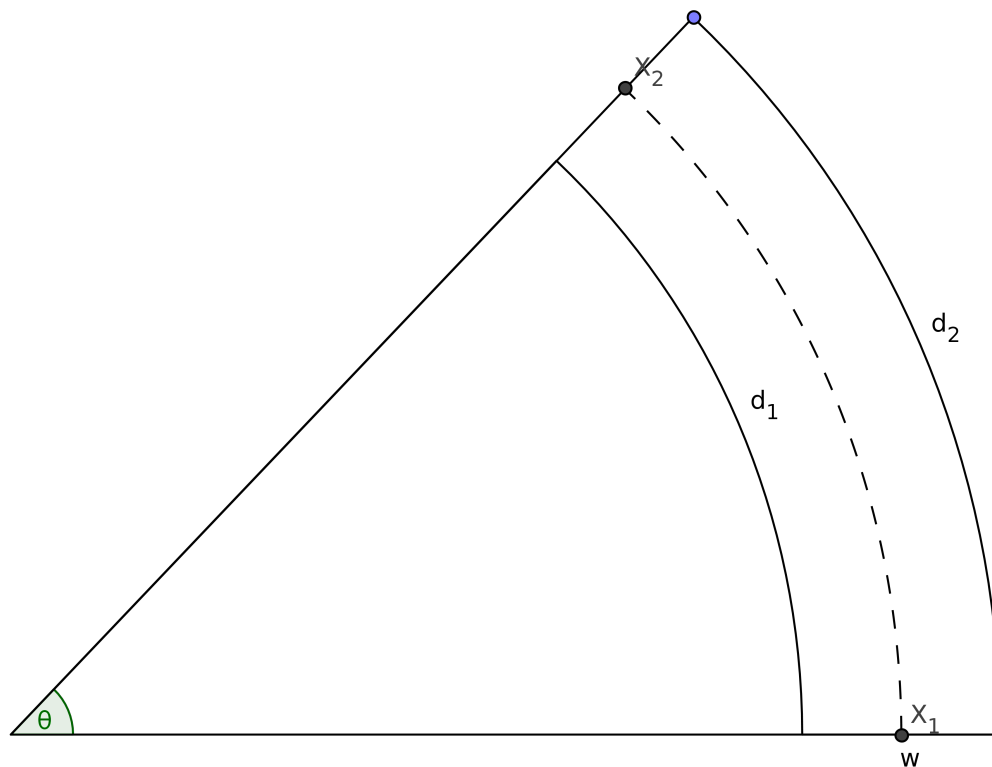
6.1.1.1 `#define HALF_PI 1.570796327`

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.

(0.77, 3.44)



(4.44, 0.63)

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

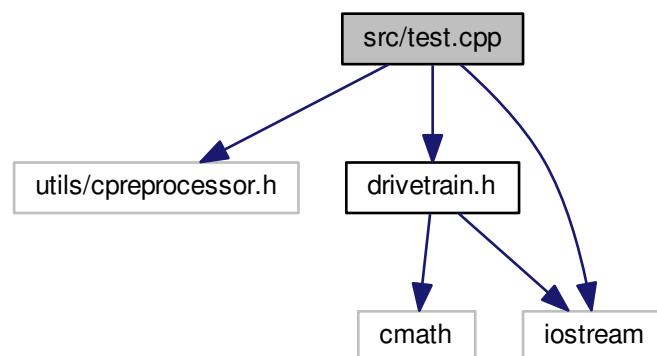
6.1.2.2 Curve operator+ (Curve *a*, Curve *b*) [inline]

Adds two curves.

6.2 src/test.cpp File Reference

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

Include dependency graph for test.cpp:



Functions

- int `main` ()

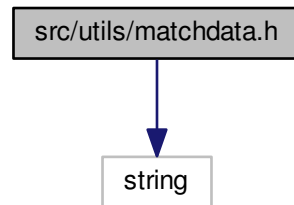
6.2.1 Function Documentation

6.2.1.1 int main ()

6.3 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

Curve, [9](#)
 heading, [9](#)
 x, [9](#)
 y, [9](#)
curveamount
 drivetrain.h, [12](#)

DoublePair, [10](#)
 u, [10](#)
 v, [10](#)

drivetrain.h
 curveamount, [12](#)
 HALF_PI, [12](#)
 operator+, [13](#)

field, [7](#)
field::side, [7](#)
 left, [7](#)
 right, [7](#)
 scale, [7](#)
 setup, [7](#)
 switch_far, [7](#)
 switch_near, [7](#)

HALF_PI
 drivetrain.h, [12](#)

heading
 Curve, [9](#)

left
 field::side, [7](#)

main
 test.cpp, [13](#)

operator+
 drivetrain.h, [13](#)

right
 field::side, [7](#)

scale
 field::side, [7](#)

setup
 field::side, [7](#)

src/drivetrain.h, [11](#)
src/test.cpp, [13](#)
src/utils/matchdata.h, [14](#)
switch_far
 field::side, [7](#)
switch_near
 field::side, [7](#)

test.cpp
 main, [13](#)

u
 DoublePair, [10](#)

v
 DoublePair, [10](#)

x
 Curve, [9](#)

y
 Curve, [9](#)