

af_bringup

Generated by Doxygen 1.8.14

Contents

1	Namespace Index	1
1.1	Packages	1
2	Data Structure Index	3
2.1	Data Structures	3
3	File Index	5
3.1	File List	5
4	Namespace Documentation	7
4.1	fuse_G_A Namespace Reference	7
4.1.1	Function Documentation	8
4.1.1.1	make_choice()	8
4.1.1.2	matrix_from_theta()	8
4.1.1.3	normalization()	8
4.1.1.4	sendTransform()	9
4.1.1.5	theta_from_matrix()	9
4.1.1.6	tran_mat44()	9
4.1.1.7	tran_theta_T()	10
4.1.2	Variable Documentation	10
4.1.2.1	amcl_pose	10
4.1.2.2	gps_pose	10
4.1.2.3	listener	10
4.1.2.4	MAX_STD	10
4.1.2.5	odom_pose	11

4.1.2.6	out_pose	11
4.1.2.7	pub	11
4.1.2.8	t1	11
4.1.2.9	t2	11
4.1.2.10	tfBuffer	11
4.2	select_A Namespace Reference	11
4.2.1	Function Documentation	12
4.2.1.1	amcl_global_update()	13
4.2.1.2	amcl_initial_callback()	13
4.2.1.3	amcl_initial_update()	13
4.2.1.4	amcl_nomotion_update()	13
4.2.1.5	amcl_process()	14
4.2.1.6	amcl_recovery()	14
4.2.1.7	angle_diff()	14
4.2.1.8	main()	14
4.2.1.9	make_choice()	14
4.2.1.10	matrix_from_theta()	15
4.2.1.11	normalize()	15
4.2.1.12	sendTransform()	15
4.2.1.13	theta_from_matrix()	16
4.2.1.14	tran_mat44()	16
4.2.1.15	tran_theta_T()	16
4.2.2	Variable Documentation	16
4.2.2.1	flag_robot_move	17
4.2.2.2	MAX_STD	17
4.2.2.3	STATIC_LIMIT_ANGULAR	17
4.2.2.4	STATIC_LIMIT_LINEAR	17
4.3	serial_af Namespace Reference	17
4.3.1	Function Documentation	18
4.3.1.1	main()	18
4.3.1.2	range_pub()	18
4.3.1.3	serial_process()	19
4.3.1.4	sound_range_init()	19
4.3.2	Variable Documentation	19
4.3.2.1	crc_novetel	20
4.3.2.2	fuse_gps_odom	20
4.3.2.3	mutex	20
4.3.2.4	sound_range	20
4.3.2.5	sound_seq	20

5 Data Structure Documentation	21
5.1 fuse_G_A.AMCL_unit Class Reference	21
5.1.1 Detailed Description	22
5.1.2 Constructor & Destructor Documentation	22
5.1.2.1 __init__()	22
5.1.3 Member Function Documentation	22
5.1.3.1 fix()	22
5.1.3.2 output()	23
5.1.3.3 update()	23
5.1.4 Field Documentation	23
5.1.4.1 _theta_base	23
5.1.4.2 _x_base	23
5.1.4.3 _y_base	24
5.1.4.4 theta	24
5.1.4.5 theta_std	24
5.1.4.6 x	24
5.1.4.7 x_std	24
5.1.4.8 y	24
5.1.4.9 y_std	25
5.2 select_A.AMCL_unit Class Reference	25
5.2.1 Detailed Description	26
5.2.2 Constructor & Destructor Documentation	26
5.2.2.1 __init__()	26
5.2.3 Member Function Documentation	26
5.2.3.1 fix()	26
5.2.3.2 lock()	26
5.2.3.3 output()	27
5.2.3.4 update()	27
5.2.4 Field Documentation	27
5.2.4.1 _theta_base	27

5.2.4.2	_x_base	27
5.2.4.3	_y_base	28
5.2.4.4	theta	28
5.2.4.5	theta_std	28
5.2.4.6	x	28
5.2.4.7	x_std	28
5.2.4.8	y	28
5.2.4.9	y_std	29
5.3	serial_af.Encoder Class Reference	29
5.3.1	Detailed Description	29
5.3.2	Constructor & Destructor Documentation	29
5.3.2.1	__init__()	29
5.3.3	Member Function Documentation	30
5.3.3.1	plot()	30
5.3.4	Field Documentation	30
5.3.4.1	encoder_time	30
5.3.4.2	left_encoder_val	30
5.3.4.3	right_encoder_val	30
5.4	serial_af.Fuse_GPS_Odom Class Reference	30
5.4.1	Detailed Description	31
5.4.2	Constructor & Destructor Documentation	31
5.4.2.1	__init__()	31
5.4.3	Member Function Documentation	31
5.4.3.1	fill()	31
5.4.4	Field Documentation	31
5.4.4.1	fuse_encoder	32
5.4.4.2	fuse_gps	32
5.4.4.3	fuse_gps_odom_msg	32
5.5	serial_af.GPS Class Reference	32
5.5.1	Detailed Description	33

5.5.2	Constructor & Destructor Documentation	33
5.5.2.1	__init__()	34
5.5.3	Member Function Documentation	34
5.5.3.1	plot()	34
5.5.4	Field Documentation	34
5.5.4.1	gps_time	34
5.5.4.2	hdg_std	34
5.5.4.3	heading	34
5.5.4.4	hgt	35
5.5.4.5	hgt_std	35
5.5.4.6	hor_spd	35
5.5.4.7	lat	35
5.5.4.8	lat_std	35
5.5.4.9	lon	35
5.5.4.10	lon_std	36
5.5.4.11	Pos_type	36
5.5.4.12	Slo_stat	36
5.5.4.13	solnSVs	36
5.5.4.14	SVs	36
5.5.4.15	Trk_gnd	36
5.5.4.16	Vert_spd	37
5.6	select_A.GPS_unit Class Reference	37
5.6.1	Detailed Description	37
5.6.2	Constructor & Destructor Documentation	38
5.6.2.1	__init__()	38
5.6.3	Member Function Documentation	38
5.6.3.1	lock()	38
5.6.3.2	output()	38
5.6.3.3	update()	38
5.6.4	Field Documentation	39

5.6.4.1	theta	39
5.6.4.2	theta_std	39
5.6.4.3	x	39
5.6.4.4	x_std	39
5.6.4.5	y	39
5.6.4.6	y_std	40
5.7	fuse_G_A.GPS_unit Class Reference	40
5.7.1	Detailed Description	40
5.7.2	Constructor & Destructor Documentation	40
5.7.2.1	__init__()	41
5.7.3	Member Function Documentation	41
5.7.3.1	output()	41
5.7.3.2	update()	41
5.7.4	Field Documentation	41
5.7.4.1	theta	41
5.7.4.2	theta_std	42
5.7.4.3	x	42
5.7.4.4	x_std	42
5.7.4.5	y	42
5.7.4.6	y_std	42
5.8	fuse_G_A.Odom_unit Class Reference	42
5.8.1	Detailed Description	43
5.8.2	Constructor & Destructor Documentation	43
5.8.2.1	__init__()	43
5.8.3	Member Function Documentation	43
5.8.3.1	output()	43
5.8.4	Field Documentation	44
5.8.4.1	theta	44
5.8.4.2	theta_std	44
5.8.4.3	x	44

5.8.4.4	<code>x_std</code>	44
5.8.4.5	<code>y</code>	44
5.8.4.6	<code>y_std</code>	45
5.9	<code>select_A.Odom_unit</code> Class Reference	45
5.9.1	Detailed Description	45
5.9.2	Constructor & Destructor Documentation	45
5.9.2.1	<code>__init__()</code>	46
5.9.3	Member Function Documentation	46
5.9.3.1	<code>output()</code>	46
5.9.4	Field Documentation	46
5.9.4.1	<code>theta</code>	46
5.9.4.2	<code>theta_std</code>	46
5.9.4.3	<code>x</code>	46
5.9.4.4	<code>x_std</code>	47
5.9.4.5	<code>y</code>	47
5.9.4.6	<code>y_std</code>	47
5.10	<code>select_A.SELECTED</code> Class Reference	47
5.10.1	Detailed Description	48
5.10.2	Constructor & Destructor Documentation	48
5.10.2.1	<code>__init__()</code>	48
5.10.3	Member Function Documentation	48
5.10.3.1	<code>output()</code>	48
5.10.3.2	<code>set_map()</code>	49
5.10.3.3	<code>set_odom()</code>	49
5.10.4	Field Documentation	49
5.10.4.1	<code>choice</code>	49
5.10.4.2	<code>fuse_tf</code>	49
5.10.4.3	<code>theta</code>	49
5.10.4.4	<code>theta_choice</code>	50
5.10.4.5	<code>x</code>	50

5.10.4.6	x_choice	50
5.10.4.7	y	50
5.10.4.8	y_choice	50
5.11	fuse_G_A.SELECTED Class Reference	50
5.11.1	Detailed Description	51
5.11.2	Constructor & Destructor Documentation	51
5.11.2.1	__init__()	51
5.11.3	Member Function Documentation	51
5.11.3.1	output()	52
5.11.3.2	set_map()	52
5.11.3.3	set_odom()	52
5.11.4	Field Documentation	52
5.11.4.1	choice	52
5.11.4.2	fuse_tf	52
5.11.4.3	theta	53
5.11.4.4	theta_choice	53
5.11.4.5	x	53
5.11.4.6	x_choice	53
5.11.4.7	y	53
5.11.4.8	y_choice	53
6	File Documentation	55
6.1	fuse_G_A.py File Reference	55
6.2	select_A.py File Reference	56
6.3	serial_af.py File Reference	57
Index		59

Chapter 1

Namespace Index

1.1 Packages

Here are the packages with brief descriptions (if available):

fuse_G_A	7
select_A	11
serial_af	17

Chapter 2

Data Structure Index

2.1 Data Structures

Here are the data structures with brief descriptions:

fuse_G_A.AMCL_unit	
AMCL存储类	21
select_A.AMCL_unit	
AMCL存储类	25
serial_af.Encoder	
类用来存储里程计信息	29
serial_af.Fuse_GPS_Odom	
Class Fuse_GPS_Odom 用来存储GPS和Encoder的信息	30
serial_af.GPS	
存储GPS信息的类	32
select_A.GPS_unit	
GPS数据存储类	37
fuse_G_A.GPS_unit	
GPS数据存储类	40
fuse_G_A.Odom_unit	
里程计数据存储类	42
select_A.Odom_unit	
里程计数据存储类	45
select_A.SELECTED	
筛选结果类	47
fuse_G_A.SELECTED	
筛选结果类	50

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

fuse_G_A.py	55
select_A.py	56
serial_af.py	57

Chapter 4

Namespace Documentation

4.1 fuse_G_A Namespace Reference

Data Structures

- class [AMCL_unit](#)
AMCL存储类
- class [GPS_unit](#)
GPS数据存储类
- class [Odom_unit](#)
里程计数据存储类
- class [SELECTED](#)
筛选结果类

Functions

- def [matrix_from_theta](#) (theta)
角度生成旋转矩阵(2x2)
- def [theta_from_matrix](#) (R)
旋转矩阵(2x2)转成角度
- def [sendTransform](#) ()
发送topic
- def [tran_mat44](#) (trans)
将ros格式中的Transform数据转为矩阵(4X4)
- def [tran_theta_T](#) (trans)
将ros格式中的Transform数据转为矩阵(2X2)
- def [normalization](#) (a, b)
归一化
- def [make_choice](#) ()
筛选函数

Variables

- int `MAX_STD` = 999999999
最大的标准差
- `tfBuffer` = `tf2_ros.Buffer()`
- `listener` = `tf2_ros.TransformListener(tfBuffer)`
- `amcl_pose` = `AMCL_unit()`
- `gps_pose` = `GPS_unit()`
- `odom_pose` = `Odom_unit()`
- `out_pose` = `SELECTED()`
- `pub` = `rospy.Publisher('/fuse/tf', Fuse_tf, queue_size=100)`
- `t2` = `threading.Thread(target=sendTransform)`
- `t1` = `threading.Thread(target=make_choice)`

4.1.1 Function Documentation

4.1.1.1 `make_choice()`

```
def fuse_G_A.make_choice ( )
```

筛选函数

4.1.1.2 `matrix_from_theta()`

```
def fuse_G_A.matrix_from_theta (
    theta )
```

角度生成旋转矩阵(2x2)

Parameters

<i>theta</i>	输入角度
--------------	------

Returns

输出旋转矩阵

4.1.1.3 `normalization()`

```
def fuse_G_A.normalization (
    a,
    b )
```

归一化

Parameters

a	输入
b	输入

Returns

a 的归一化值

b 的归一化值

4.1.1.4 sendTransform()

```
def fuse_G_A.sendTransform ( )
```

发送topic

40hz发布topic

Parameters

out_pose	ros输出格式数据
-------------	-----------

4.1.1.5 theta_from_matrix()

```
def fuse_G_A.theta_from_matrix (
     $R$  )
```

旋转矩阵(2x2)转成角度

Parameters

R	输入旋转矩阵
-----	--------

Returns

输出角度值

4.1.1.6 tran_mat44()

```
def fuse_G_A.tran_mat44 (
     $trans$  )
```

将ros格式中的Transform数据转为矩阵(4X4)

Parameters

<i>trans</i>	ros格式Transform数据
--------------	------------------

4.1.1.7 tran_theta_T()

```
def fuse_G_A.tran_theta_T (
    trans )
```

将ros格式中的Transform数据转为矩阵(2X2)

Parameters

<i>trans</i>	ros格式Transform数据
--------------	------------------

4.1.2 Variable Documentation**4.1.2.1 amcl_pose**

```
fuse_G_A.amcl_pose = AMCL_unit()
```

4.1.2.2 gps_pose

```
fuse_G_A.gps_pose = GPS_unit()
```

4.1.2.3 listener

```
fuse_G_A.listener = tf2_ros.TransformListener(tfBuffer)
```

4.1.2.4 MAX_STD

```
int fuse_G_A.MAX_STD = 999999999
```

最大的标准差

4.1.2.5 odom_pose

```
fuse_G_A.odom_pose = Odom_unit()
```

4.1.2.6 out_pose

```
fuse_G_A.out_pose = SELECTED()
```

4.1.2.7 pub

```
fuse_G_A.pub = rospy.Publisher('/fuse/tf', Fuse_tf, queue_size=100)
```

4.1.2.8 t1

```
fuse_G_A.t1 = threading.Thread(target=make_choice)
```

4.1.2.9 t2

```
fuse_G_A.t2 = threading.Thread(target=sendTransform)
```

4.1.2.10 tfBuffer

```
fuse_G_A.tfBuffer = tf2_ros.Buffer()
```

4.2 select_A Namespace Reference

Data Structures

- class [AMCL_unit](#)
*AMCL*存储类
- class [GPS_unit](#)
*GPS*数据存储类
- class [Odom_unit](#)
里程计数据存储类
- class [SELECTED](#)
筛选结果类

Functions

- def `normalize` (z)
归一化
- def `angle_diff` (a, b)
角度差值
- def `matrix_from_theta` (theta)
角度生成旋转矩阵(2x2)
- def `theta_from_matrix` (R)
旋转矩阵(2x2)转成角度
- def `sendTransform` (out_pose)
发送TF(topic)
- def `tran_mat44` (trans)
将ros格式中的Transform数据转为矩阵(4X4)
- def `tran_theta_T` (trans)
将ros格式中的Transform数据转为矩阵(2X2)
- def `make_choice` ()
筛选函数
- def `amcl_initial_update` (x_std=1, y_std=1, theta_std=1)
AMCL初始化函数
- def `amcl_initial_callback` (msg)
AMCL 初始化回调函数
- def `amcl_global_update` ()
AMCL全局撒粒子
- def `amcl_nomotion_update` ()
AMCL未运动状态下粒子更新
- def `amcl_recovery` ()
AMCL恢复状态过程
- def `amcl_process` ()
AMCL线程
- def `main` ()
主函数

Variables

- int `MAX_STD` = 999999999
最大的标准差
- int `flag_robot_move` = 0
机器人是否运动标志
- float `STATIC_LIMIT_LINEAR` = 0.05
判断机器人是否运动的速度限制
- float `STATIC_LIMIT_ANGULAR` = 0.05
判断机器人是否运动的角度限制

4.2.1 Function Documentation

4.2.1.1 amcl_global_update()

```
def select_A.amcl_global_update ( )
```

AMCL全局撒粒子

4.2.1.2 amcl_initial_callback()

```
def select_A.amcl_initial_callback (
    msg )
```

AMCL 初始化回调函数

Parameters

<i>msg</i>	ros回调数据
------------	---------

4.2.1.3 amcl_initial_update()

```
def select_A.amcl_initial_update (
    x_std = 1,
    y_std = 1,
    theta_std = 1 )
```

AMCL初始化函数

Parameters

<i>x_std</i>	x方向方差
<i>y_std</i>	y方向方差
<i>theta_std</i>	角度方差

4.2.1.4 amcl_nomotion_update()

```
def select_A.amcl_nomotion_update ( )
```

AMCL未运动状态下粒子更新

4.2.1.5 amcl_process()

```
def select_A.amcl_process ( )
```

AMCL线程

包括AMCL位置纠正,AMCL位置正确度判断和AMCL自恢复

4.2.1.6 amcl_recovery()

```
def select_A.amcl_recovery ( )
```

AMCL恢复状态过程

4.2.1.7 angle_diff()

```
def select_A.angle_diff (
    a,
    b )
```

角度差值

Parameters

<i>a</i>	输入角度
<i>b</i>	输入角度

Returns

角度差值

4.2.1.8 main()

```
def select_A.main ( )
```

主函数

4.2.1.9 make_choice()

```
def select_A.make_choice ( )
```

筛选函数

4.2.1.10 matrix_from_theta()

```
def select_A.matrix_from_theta (  
    theta )
```

角度生成旋转矩阵(2x2)

Parameters

<i>theta</i>	输入角度
--------------	------

Returns

输出旋转矩阵

4.2.1.11 normalize()

```
def select_A.normalize (  
    z )
```

归一化

Parameters

<i>z</i>	角度输入
----------	------

Returns

角度输出(0~pi)

4.2.1.12 sendTransform()

```
def select_A.sendTransform (  
    out_pose )
```

发送TF(topic)

Parameters

<i>out_pose</i>	ros输出格式数据
-----------------	-----------

4.2.1.13 `theta_from_matrix()`

```
def select_A.theta_from_matrix (
    R )
```

旋转矩阵(2x2)转成角度

Parameters

<i>R</i>	输入旋转矩阵
----------	--------

Returns

输出角度值

4.2.1.14 `tran_mat44()`

```
def select_A.tran_mat44 (
    trans )
```

将ros格式中的Transform数据转为矩阵(4X4)

Parameters

<i>trans</i>	ros格式Transform数据
--------------	------------------

4.2.1.15 `tran_theta_T()`

```
def select_A.tran_theta_T (
    trans )
```

将ros格式中的Transform数据转为矩阵(2X2)

Parameters

<i>trans</i>	ros格式Transform数据
--------------	------------------

4.2.2 Variable Documentation

4.2.2.1 flag_robot_move

```
int select_A.flag_robot_move = 0
```

机器人是否运动标志

4.2.2.2 MAX_STD

```
int select_A.MAX_STD = 999999999
```

最大的标准差

4.2.2.3 STATIC_LIMIT_ANGULAR

```
float select_A.STATIC_LIMIT_ANGULAR = 0.05
```

判断机器人是否运动的角度限制

4.2.2.4 STATIC_LIMIT_LINEAR

```
float select_A.STATIC_LIMIT_LINEAR = 0.05
```

判断机器人是否运动的速度限制

4.3 serial_af Namespace Reference

Data Structures

- class [Encoder](#)
类用来存储里程计信息
- class [Fuse_GPS_Odom](#)
class Fuse_GPS_Odom 用来存储GPS和Encoder的信息
- class [GPS](#)
存储GPS信息的类

Functions

- def [sound_range_init](#) ()
超声波数据初始化
- def [range_pub](#) (range_msg)
range_pub range的激光输出函数
- def [serial_process](#) (port='/dev/pts/27', baudrate=460800, timeout=1)
serial_process 串口线程
- def [main](#) ()
*main*函数

Variables

- `crc_novetal` = `crcmod.mkCrcFun(0x104C11DB7, 0, True, 0)`
crc32校验程序(GPS)
- `mutex` = `threading.Lock()`
线程锁
- `fuse_gps_odom` = `Fuse_GPS_Odom()`
- `sound_range` = `LaserScan()`
超声波模拟激光输出类
- `int sound_seq` = 0
超声波输出计数

4.3.1 Function Documentation

4.3.1.1 `main()`

```
def serial_af.main ( )
```

`main`函数

`ros`的初始化 `serial`线程的开始 `spin`线程的开始

4.3.1.2 `range_pub()`

```
def serial_af.range_pub (
    range_msg )
```

`range_pub` `range`的激光输出函数

将`list`格式的`range`数据转发成`ros_laser`模式,并通过`/range/*` topic发出

Parameters

<code>range_msg</code>	<code>range</code> 的输入[1,...,3](list)
------------------------	---------------------------------------

Examples

```
>>> a = float("inf")
inf
>>> b = float("inf")
inf
>>> a + b
inf
>>> a * b
inf
>>> a + 12
inf
>>> a - 12
inf
```

4.3.1.3 serial_process()

```
def serial_af.serial_process (
    port = '/dev/pts/27',
    baudrate = 460800,
    timeout = 1 )
```

serial_process 串口线程

处理串口中的三包数据,GPS,里程计和超声波,GPS和里程计分别存储在GPS类和Encoder类中,在里程计接收过程中讲两个类的数据存储在融合类Fuse_GPS_Odom中.

Parameters

<i>port</i>	串口
<i>baudrate</i>	波特率
<i>timeout</i>	时间容差

See also

ser_1 读取的第一个串口值
encoder_a 串口剩余部分
encoder_b 解的串口数据
crc 解的串口校验值
encoder_crc 计算的串口校验值

Notes

GPS和超声波数据类似里程计读取流程

4.3.1.4 sound_range_init()

```
def serial_af.sound_range_init ( )
```

超声波数据初始化

See also

range_max 超声波最远距离
range_min 超声波最小距离(本来是0.290),range值小于等于最小值,输出无效,固最小值缩小
scan_time 超声波扫描时间
time_increment 超声波模拟激光模拟数据
angle_increment 超声波模拟激光模拟数据
angle_min 超声波的探测角($\pm 30^\circ$)
angle_max 超声波的探测角($\pm 30^\circ$)

4.3.2 Variable Documentation

4.3.2.1 `crc_novetal`

```
serial_af.crc_novetal = crcmod.mkCrcFun(0x104C11DB7, 0, True, 0)
```

`crc32`校验程序(GPS)

4.3.2.2 `fuse_gps_odom`

```
serial_af.fuse_gps_odom = Fuse_GPS_Odom()
```

4.3.2.3 `mutex`

```
serial_af.mutex = threading.Lock()
```

线程锁

4.3.2.4 `sound_range`

```
serial_af.sound_range = LaserScan()
```

超声波模拟激光输出类

4.3.2.5 `sound_seq`

```
int serial_af.sound_seq = 0
```

超声波输出计数

Chapter 5

Data Structure Documentation

5.1 fuse_G_A.AMCL_unit Class Reference

AMCL存储类

Public Member Functions

- def `__init__` (self)
构造函数
- def `update` (self, odom_msg)
里程计更新
- def `fix` (self, odom_map_msg)
AMCL修正输入
- def `output` (self)
输出

Data Fields

- `x`
x
- `x_std`
x标准差
- `y`
y
- `y_std`
y标准差
- `theta`
角度
- `theta_std`
角度标准差

Private Attributes

- [_theta_base](#)
累计角度
- [_x_base](#)
累计x
- [_y_base](#)
累计y

5.1.1 Detailed Description

AMCL存储类

5.1.2 Constructor & Destructor Documentation

5.1.2.1 __init__()

```
def fuse_G_A.AMCL_unit.__init__ (
    self )
```

构造函数

5.1.3 Member Function Documentation

5.1.3.1 fix()

```
def fuse_G_A.AMCL_unit.fix (
    self,
    odom_map_msg )
```

AMCL修正输入

将AMCL修正输入的map->odom和里程计输入融合生成实时的AMCL输出值(机器人在地图上的激光定位)

Parameters

<i>odom_map_msg</i>	AMCL的输入的map->odom
---------------------	-------------------

5.1.3.2 output()

```
def fuse_G_A.AMCL_unit.output (
    self )
```

输出

5.1.3.3 update()

```
def fuse_G_A.AMCL_unit.update (
    self,
    odom_msg )
```

里程计更新

Parameters

<i>odom_msg</i>	ros里程计输入
-----------------	----------

See also

[_pos](#) 位置矩阵
[_pose_base](#) 位置基准矩阵
[_pose_update](#) 坐标系变换
[_quat](#) 四元数
[flag_robot_move](#) 机器人运动标志

5.1.4 Field Documentation

5.1.4.1 _theta_base

```
fuse_G_A.AMCL_unit._theta_base [private]
```

累计角度

5.1.4.2 _x_base

```
fuse_G_A.AMCL_unit._x_base [private]
```

累计x

5.1.4.3 `_y_base`

`fuse_G_A.AMCL_unit._y_base` [private]

累计y

5.1.4.4 `theta`

`fuse_G_A.AMCL_unit.theta`

角度

5.1.4.5 `theta_std`

`fuse_G_A.AMCL_unit.theta_std`

角度标准差

5.1.4.6 `x`

`fuse_G_A.AMCL_unit.x`

x

5.1.4.7 `x_std`

`fuse_G_A.AMCL_unit.x_std`

x标准差

5.1.4.8 `y`

`fuse_G_A.AMCL_unit.y`

y

5.1.4.9 y_std

```
fuse_G_A.AMCL_unit.y_std
```

y标准差

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

- [fuse_G_A.py](#)

5.2 select_A.AMCL_unit Class Reference

AMCL存储类

Public Member Functions

- [def __init__](#) (self)
构造函数
- [def update](#) (self, odom_msg)
里程计更新
- [def fix](#) (self, odom_map_msg)
AMCL修正输入
- [def lock](#) (self)
将AMCL输出结果置为不可信
- [def output](#) (self)
输出

Data Fields

- [x](#)
x
- [x_std](#)
x标准差
- [y](#)
y
- [y_std](#)
y标准差
- [theta](#)
角度
- [theta_std](#)
角度标准差

Private Attributes

- [_theta_base](#)
累计角度
- [_x_base](#)
累计x
- [_y_base](#)
累计y

5.2.1 Detailed Description

AMCL存储类

5.2.2 Constructor & Destructor Documentation

5.2.2.1 __init__()

```
def select_A.AMCL_unit.__init__ (
    self )
```

构造函数

5.2.3 Member Function Documentation

5.2.3.1 fix()

```
def select_A.AMCL_unit.fix (
    self,
    odom_map_msg )
```

AMCL修正输入

将AMCL修正输入的map->odom和里程计输入融合生成实时的AMCL输出值(机器人在地图上的激光定位)

Parameters

<i>odom_map_msg</i>	AMCL的输入的map->odom
---------------------	-------------------

5.2.3.2 lock()

```
def select_A.AMCL_unit.lock (
    self )
```

将AMCL输出结果置为不可信

5.2.3.3 output()

```
def select_A.AMCL_unit.output (
    self )
```

输出

5.2.3.4 update()

```
def select_A.AMCL_unit.update (
    self,
    odom_msg )
```

里程计更新

Parameters

<code>odom_msg</code>	ros里程计输入
-----------------------	----------

See also

`_pos` 位置矩阵
`_pose_base` 位置基准矩阵
`_pose_update` 坐标系变换
`_quat` 四元数
[flag_robot_move](#) 机器人运动标志

5.2.4 Field Documentation

5.2.4.1 _theta_base

```
select_A.AMCL_unit._theta_base [private]
```

累计角度

5.2.4.2 _x_base

```
select_A.AMCL_unit._x_base [private]
```

累计x

5.2.4.3 `_y_base`

`select_A.AMCL_unit._y_base` [private]

累计y

5.2.4.4 `theta`

`select_A.AMCL_unit.theta`

角度

5.2.4.5 `theta_std`

`select_A.AMCL_unit.theta_std`

角度标准差

5.2.4.6 `x`

`select_A.AMCL_unit.x`

x

5.2.4.7 `x_std`

`select_A.AMCL_unit.x_std`

x标准差

5.2.4.8 `y`

`select_A.AMCL_unit.y`

y

5.2.4.9 y_std

`select_A.AMCL_unit.y_std`

y标准差

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

- [select_A.py](#)

5.3 serial_af.Encoder Class Reference

类用来存储里程计信息

Public Member Functions

- `def __init__ (self)`
构造函数
- `def plot (self)`
输出

Data Fields

- `left_encoder_val`
左里程计值
- `right_encoder_val`
右里程计值
- `encoder_time`
同步里程计时间

5.3.1 Detailed Description

类用来存储里程计信息

5.3.2 Constructor & Destructor Documentation

5.3.2.1 `__init__()`

```
def serial_af.Encoder.__init__ (  
    self )
```

构造函数

5.3.3 Member Function Documentation

5.3.3.1 plot()

```
def serial_af.Encoder.plot (
    self )
```

输出

5.3.4 Field Documentation

5.3.4.1 encoder_time

```
serial_af.Encoder.encoder_time
```

同步里程计时间

5.3.4.2 left_encoder_val

```
serial_af.Encoder.left_encoder_val
```

左里程计值

5.3.4.3 right_encoder_val

```
serial_af.Encoder.right_encoder_val
```

右里程计值

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

- [serial_af.py](#)

5.4 serial_af.Fuse_GPS_Odom Class Reference

class [Fuse_GPS_Odom](#) 用来存储GPS和Encoder的信息

Public Member Functions

- def `__init__` (self)
构造函数
- def `fill` (self)
将GPS和Encoder的信息添加到类中

Data Fields

- `fuse_gps`
GPS类存储值
- `fuse_encoder`
Encoder类存储值
- `fuse_gps_odom_msg`
本类融合两种存储值

5.4.1 Detailed Description

class `Fuse_GPS_Odom` 用来存储GPS和Encoder的信息

5.4.2 Constructor & Destructor Documentation

5.4.2.1 `__init__()`

```
def serial_af.Fuse_GPS_Odom.__init__ (
    self )
```

构造函数

5.4.3 Member Function Documentation

5.4.3.1 `fill()`

```
def serial_af.Fuse_GPS_Odom.fill (
    self )
```

将GPS和Encoder的信息添加到类中

5.4.4 Field Documentation

5.4.4.1 fuse_encoder

`serial_af.Fuse_GPS_Odom.fuse_encoder`

Encoder类存储值

5.4.4.2 fuse_gps

`serial_af.Fuse_GPS_Odom.fuse_gps`

GPS类存储值

5.4.4.3 fuse_gps_odom_msg

`serial_af.Fuse_GPS_Odom.fuse_gps_odom_msg`

本类融合两种存储值

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

- [serial_af.py](#)

5.5 serial_af.GPS Class Reference

存储GPS信息的类

Public Member Functions

- `def __init__ (self)`
构造函数
- `def plot (self)`
输出

Data Fields

- [Slo_stat](#)
卫星状态
- [Pos_type](#)
定位类型
- [lat](#)
纬度
- [lat_std](#)
纬度标准差
- [lon](#)
经度
- [lon_std](#)
经度标准差
- [hgt](#)
高度
- [hgt_std](#)
高度标准差
- [SVs](#)
卫星数
- [solnSVs](#)
参与解算卫星数
- [heading](#)
航向
- [hdg_std](#)
航向标准差
- [hor_spd](#)

- [Trk_gnd](#)

- [Vert_spd](#)

- [gps_time](#)
*gps*校准时间

5.5.1 Detailed Description

存储GPS信息的类

5.5.2 Constructor & Destructor Documentation

5.5.2.1 `__init__()`

```
def serial_af.GPS.__init__ (
    self )
```

构造函数

5.5.3 Member Function Documentation

5.5.3.1 `plot()`

```
def serial_af.GPS.plot (
    self )
```

输出

5.5.4 Field Documentation

5.5.4.1 `gps_time`

```
serial_af.GPS.gps_time
```

gps校准时间

5.5.4.2 `hdg_std`

```
serial_af.GPS.hdg_std
```

航向标准差

5.5.4.3 `heading`

```
serial_af.GPS.heading
```

航向

5.5.4.4 hgt

`serial_af.GPS.hgt`

高度

5.5.4.5 hgt_std

`serial_af.GPS.hgt_std`

高度标准差

5.5.4.6 hor_spd

`serial_af.GPS.hor_spd`

5.5.4.7 lat

`serial_af.GPS.lat`

纬度

5.5.4.8 lat_std

`serial_af.GPS.lat_std`

纬度标准差

5.5.4.9 lon

`serial_af.GPS.lon`

经度

5.5.4.10 lon_std

```
serial_af.GPS.lon_std
```

经度标准差

5.5.4.11 Pos_type

```
serial_af.GPS.Pos_type
```

定位类型

5.5.4.12 Slo_stat

```
serial_af.GPS.Slo_stat
```

卫星状态

5.5.4.13 solnSVs

```
serial_af.GPS.solnSVs
```

参与解算卫星数

5.5.4.14 SVs

```
serial_af.GPS.SVs
```

卫星数

5.5.4.15 Trk_gnd

```
serial_af.GPS.Trk_gnd
```

5.5.4.16 Vert_spd

`serial_af.GPS.Vert_spd`

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

- [serial_af.py](#)

5.6 select_A.GPS_unit Class Reference

GPS数据存储类

Public Member Functions

- `def __init__(self)`
构造函数
- `def update(self, gps_msg)`
GPS数据更新函数
- `def lock(self)`
将GPS输出结果置为不可信
- `def output(self)`
输出

Data Fields

- `x`
x
- `x_std`
x标准差
- `y`
y
- `y_std`
y标准差
- `theta`
角度
- `theta_std`
角度标准差

5.6.1 Detailed Description

GPS数据存储类

5.6.2 Constructor & Destructor Documentation

5.6.2.1 `__init__()`

```
def select_A.GPS_unit.__init__ (
    self )
```

构造函数

5.6.3 Member Function Documentation

5.6.3.1 `lock()`

```
def select_A.GPS_unit.lock (
    self )
```

将GPS输出结果置为不可信

5.6.3.2 `output()`

```
def select_A.GPS_unit.output (
    self )
```

输出

5.6.3.3 `update()`

```
def select_A.GPS_unit.update (
    self,
    gps_msg )
```

GPS数据更新函数

Parameters

<i>gps_msg</i>	GPS数据输入
----------------	---------

5.6.4 Field Documentation

5.6.4.1 theta

```
select_A.GPS_unit.theta
```

角度

5.6.4.2 theta_std

```
select_A.GPS_unit.theta_std
```

角度标准差

5.6.4.3 x

```
select_A.GPS_unit.x
```

x

5.6.4.4 x_std

```
select_A.GPS_unit.x_std
```

x标准差

5.6.4.5 y

```
select_A.GPS_unit.y
```

y

5.6.4.6 y_std

`select_A.GPS_unit.y_std`

y标准差

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

- [select_A.py](#)

5.7 fuse_G_A.GPS_unit Class Reference

GPS数据存储类

Public Member Functions

- [def __init__](#) (self)
构造函数
- [def update](#) (self, gps_msg)
GPS数据更新函数
- [def output](#) (self)
输出

Data Fields

- [x](#)
x
- [x_std](#)
x标准差
- [y](#)
y
- [y_std](#)
y标准差
- [theta](#)
角度
- [theta_std](#)
角度标准差

5.7.1 Detailed Description

GPS数据存储类

5.7.2 Constructor & Destructor Documentation

5.7.2.1 __init__()

```
def fuse_G_A.GPS_unit.__init__ (
    self )
```

构造函数

5.7.3 Member Function Documentation

5.7.3.1 output()

```
def fuse_G_A.GPS_unit.output (
    self )
```

输出

5.7.3.2 update()

```
def fuse_G_A.GPS_unit.update (
    self,
    gps_msg )
```

GPS数据更新函数

Parameters

<i>gps_msg</i>	GPS数据输入
----------------	---------

5.7.4 Field Documentation

5.7.4.1 theta

```
fuse_G_A.GPS_unit.theta
```

角度

5.7.4.2 theta_std

`fuse_G_A.GPS_unit.theta_std`

角度标准差

5.7.4.3 x

`fuse_G_A.GPS_unit.x`

x

5.7.4.4 x_std

`fuse_G_A.GPS_unit.x_std`

x标准差

5.7.4.5 y

`fuse_G_A.GPS_unit.y`

y

5.7.4.6 y_std

`fuse_G_A.GPS_unit.y_std`

y标准差

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

- [fuse_G_A.py](#)

5.8 fuse_G_A.Odom_unit Class Reference

里程计数据存储类

Public Member Functions

- `def __init__ (self)`
构造函数
- `def output (self)`
输出

Data Fields

- `x`
`x`
- `x_std`
`x`标准差
- `y`
`y`
- `y_std`
`y`标准差
- `theta`
角度
- `theta_std`
角度标准差

5.8.1 Detailed Description

里程计数据存储类

5.8.2 Constructor & Destructor Documentation

5.8.2.1 __init__()

```
def fuse_G_A.Odom_unit.__init__ (  
    self )
```

构造函数

5.8.3 Member Function Documentation

5.8.3.1 output()

```
def fuse_G_A.Odom_unit.output (  
    self )
```

输出

5.8.4 Field Documentation

5.8.4.1 theta

`fuse_G_A.Odom_unit.theta`

角度

5.8.4.2 theta_std

`fuse_G_A.Odom_unit.theta_std`

角度标准差

5.8.4.3 x

`fuse_G_A.Odom_unit.x`

x

5.8.4.4 x_std

`fuse_G_A.Odom_unit.x_std`

x标准差

5.8.4.5 y

`fuse_G_A.Odom_unit.y`

y

5.8.4.6 y_std

`fuse_G_A.Odom_unit.y_std`

y标准差

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

- [fuse_G_A.py](#)

5.9 select_A.Odom_unit Class Reference

里程计数据存储类

Public Member Functions

- `def __init__(self)`
构造函数
- `def output(self)`
输出

Data Fields

- [x](#)
x
- [x_std](#)
x标准差
- [y](#)
y
- [y_std](#)
y标准差
- [theta](#)
角度
- [theta_std](#)
角度标准差

5.9.1 Detailed Description

里程计数据存储类

5.9.2 Constructor & Destructor Documentation

5.9.2.1 `__init__()`

```
def select_A.Odom_unit.__init__ (
    self )
```

构造函数

5.9.3 Member Function Documentation

5.9.3.1 `output()`

```
def select_A.Odom_unit.output (
    self )
```

输出

5.9.4 Field Documentation

5.9.4.1 `theta`

```
select_A.Odom_unit.theta
```

角度

5.9.4.2 `theta_std`

```
select_A.Odom_unit.theta_std
```

角度标准差

5.9.4.3 `x`

```
select_A.Odom_unit.x
```

x

5.9.4.4 x_std

```
select_A.Odom_unit.x_std
```

x标准差

5.9.4.5 y

```
select_A.Odom_unit.y
```

y

5.9.4.6 y_std

```
select_A.Odom_unit.y_std
```

y标准差

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

- [select_A.py](#)

5.10 select_A.SELECTED Class Reference

筛选结果类

Public Member Functions

- `def __init__ (self)`
构造函数
- `def set_map (self, x, y, theta)`
将选则的最终结果数据放在输出中
- `def set_odom (self, x, y, theta)`
将里程计数据放在输出中
- `def output (self)`
输出

Data Fields

- [choice](#)
- [x](#)
- x*
- [x_choice](#)
- [y](#)
- y*
- [y_choice](#)
- [theta](#)
- 角度
- [theta_choice](#)
- [fuse_tf](#)
- ros*输出格式

5.10.1 Detailed Description

筛选结果类

5.10.2 Constructor & Destructor Documentation

5.10.2.1 `__init__()`

```
def select_A.SELECTED.__init__ (
    self )
```

构造函数

5.10.3 Member Function Documentation

5.10.3.1 `output()`

```
def select_A.SELECTED.output (
    self )
```

输出

5.10.3.2 set_map()

```
def select_A.SELECTED.set_map (
    self,
    x,
    y,
    theta )
```

将选则的最终结果数据放在输出中

5.10.3.3 set_odom()

```
def select_A.SELECTED.set_odom (
    self,
    x,
    y,
    theta )
```

将里程计数据放在输出中

5.10.4 Field Documentation

5.10.4.1 choice

```
select_A.SELECTED.choice
```

5.10.4.2 fuse_tf

```
select_A.SELECTED.fuse_tf
```

ros输出格式

5.10.4.3 theta

```
select_A.SELECTED.theta
```

角度

5.10.4.4 `theta_choice`

`select_A.SELECTED.theta_choice`

5.10.4.5 `x`

`select_A.SELECTED.x`

`x`

5.10.4.6 `x_choice`

`select_A.SELECTED.x_choice`

5.10.4.7 `y`

`select_A.SELECTED.y`

`y`

5.10.4.8 `y_choice`

`select_A.SELECTED.y_choice`

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

- [select_A.py](#)

5.11 `fuse_G_A.SELECTED` Class Reference

筛选结果类

Public Member Functions

- `def __init__ (self)`
构造函数
- `def set_map (self, x, y, theta)`
将选则的最终结果数据放在输出中
- `def set_odom (self, x, y, theta)`
将里程计数据放在输出中
- `def output (self)`
输出

Data Fields

- [choice](#)
- [x](#)
 x
- [x_choice](#)
- [y](#)
 y
- [y_choice](#)
- [theta](#)
角度
- [theta_choice](#)
- [fuse_tf](#)
*ros*输出格式

5.11.1 Detailed Description

筛选结果类

5.11.2 Constructor & Destructor Documentation

5.11.2.1 `__init__()`

```
def fuse_G_A.SELECTED.__init__ (
    self )
```

构造函数

5.11.3 Member Function Documentation

5.11.3.1 output()

```
def fuse_G_A.SELECTED.output (
    self )
```

输出

5.11.3.2 set_map()

```
def fuse_G_A.SELECTED.set_map (
    self,
    x,
    y,
    theta )
```

将选则的最终结果数据放在输出中

5.11.3.3 set_odom()

```
def fuse_G_A.SELECTED.set_odom (
    self,
    x,
    y,
    theta )
```

将里程计数据放在输出中

5.11.4 Field Documentation

5.11.4.1 choice

```
fuse_G_A.SELECTED.choice
```

5.11.4.2 fuse_tf

```
fuse_G_A.SELECTED.fuse_tf
```

ros输出格式

5.11.4.3 theta

`fuse_G_A.SELECTED.theta`

角度

5.11.4.4 theta_choice

`fuse_G_A.SELECTED.theta_choice`

5.11.4.5 x

`fuse_G_A.SELECTED.x`

x

5.11.4.6 x_choice

`fuse_G_A.SELECTED.x_choice`

5.11.4.7 y

`fuse_G_A.SELECTED.y`

y

5.11.4.8 y_choice

`fuse_G_A.SELECTED.y_choice`

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

- [fuse_G_A.py](#)

Chapter 6

File Documentation

6.1 fuse_G_A.py File Reference

Data Structures

- class [fuse_G_A.AMCL_unit](#)
*AMCL*存储类
- class [fuse_G_A.Odom_unit](#)
里程计数据存储类
- class [fuse_G_A.GPS_unit](#)
*GPS*数据存储类
- class [fuse_G_A.SELECTED](#)
筛选结果类

Namespaces

- [fuse_G_A](#)

Functions

- def [fuse_G_A.matrix_from_theta](#) (theta)
角度生成旋转矩阵(2x2)
- def [fuse_G_A.theta_from_matrix](#) (R)
旋转矩阵(2x2)转成角度
- def [fuse_G_A.sendTransform](#) ()
发送*topic*
- def [fuse_G_A.tran_mat44](#) (trans)
将*ros*格式中的*Transform*数据转为矩阵(4X4)
- def [fuse_G_A.tran_theta_T](#) (trans)
将*ros*格式中的*Transform*数据转为矩阵(2X2)
- def [fuse_G_A.normalization](#) (a, b)
归一化
- def [fuse_G_A.make_choice](#) ()
筛选函数

Variables

- `int fuse_G_A.MAX_STD = 999999999`
最大的标准差
- `tf2_ros.Buffer()`
- `tf2_ros.TransformListener(tfBuffer)`
- `AMCL_unit()`
- `GPS_unit()`
- `Odom_unit()`
- `SELECTED()`
- `rospy.Publisher('/fuse/tf', Fuse_tf, queue_size=100)`
- `threading.Thread(target=sendTransform)`
- `threading.Thread(target=make_choice)`

6.2 select_A.py File Reference

Data Structures

- class `select_A.AMCL_unit`
*AMCL*存储类
- class `select_A.Odom_unit`
里程计数据存储类
- class `select_A.GPS_unit`
*GPS*数据存储类
- class `select_A.SELECTED`
筛选结果类

Namespaces

- `select_A`

Functions

- def `select_A.normalize (z)`
归一化
- def `select_A.angle_diff (a, b)`
角度差值
- def `select_A.matrix_from_theta (theta)`
角度生成旋转矩阵(2x2)
- def `select_A.theta_from_matrix (R)`
旋转矩阵(2x2)转成角度
- def `select_A.sendTransform (out_pose)`
发送*TF(topic)*
- def `select_A.tran_mat44 (trans)`
将*ros*格式中的*Transform*数据转为矩阵(4X4)
- def `select_A.tran_theta_T (trans)`
将*ros*格式中的*Transform*数据转为矩阵(2X2)
- def `select_A.make_choice ()`

- 筛选函数
- def [select_A.amcl_initial_update](#) (x_std=1, y_std=1, theta_std=1)
AMCL初始化函数
- def [select_A.amcl_initial_callback](#) (msg)
AMCL 初始化回调函数
- def [select_A.amcl_global_update](#) ()
AMCL全局撒粒子
- def [select_A.amcl_nomotion_update](#) ()
AMCL未运动状态下粒子更新
- def [select_A.amcl_recovery](#) ()
AMCL恢复状态过程
- def [select_A.amcl_process](#) ()
AMCL线程
- def [select_A.main](#) ()
主函数

Variables

- int [select_A.MAX_STD](#) = 999999999
最大的标准差
- int [select_A.flag_robot_move](#) = 0
机器人是否运动标志
- float [select_A.STATIC_LIMIT_LINEAR](#) = 0.05
判断机器人是否运动的速度限制
- float [select_A.STATIC_LIMIT_ANGULAR](#) = 0.05
判断机器人是否运动的角度限制

6.3 serial_af.py File Reference

Data Structures

- class [serial_af.Fuse_GPS_Odom](#)
class Fuse_GPS_Odom 用来存储GPS和Encoder的信息
- class [serial_af.GPS](#)
存储GPS信息的类
- class [serial_af.Encoder](#)
类用来存储里程计信息

Namespaces

- [serial_af](#)

Functions

- def `serial_af.sound_range_init` ()
超声波数据初始化
- def `serial_af.range_pub` (range_msg)
range_pub *range*的激光输出函数
- def `serial_af.serial_process` (port='/dev/pts/27', baudrate=460800, timeout=1)
serial_process 串口线程
- def `serial_af.main` ()
*main*函数

Variables

- `serial_af.crc_novetal` = `crcmod.mkCrcFun(0x104C11DB7, 0, True, 0)`
*crc32*校验程序(*GPS*)
- `serial_af.mutex` = `threading.Lock()`
线程锁
- `serial_af.fuse_gps_odom` = `Fuse_GPS_Odom()`
- `serial_af.sound_range` = `LaserScan()`
超声波模拟激光输出类
- int `serial_af.sound_seq` = 0
超声波输出计数

Index

- `__init__`
 - `fuse_G_A::AMCL_unit`, [22](#)
 - `fuse_G_A::GPS_unit`, [40](#)
 - `fuse_G_A::Odom_unit`, [43](#)
 - `fuse_G_A::SELECTED`, [51](#)
 - `select_A::AMCL_unit`, [26](#)
 - `select_A::GPS_unit`, [38](#)
 - `select_A::Odom_unit`, [45](#)
 - `select_A::SELECTED`, [48](#)
 - `serial_af::Encoder`, [29](#)
 - `serial_af::Fuse_GPS_Odom`, [31](#)
 - `serial_af::GPS`, [33](#)
 - `_theta_base`
 - `fuse_G_A::AMCL_unit`, [23](#)
 - `select_A::AMCL_unit`, [27](#)
 - `_x_base`
 - `fuse_G_A::AMCL_unit`, [23](#)
 - `select_A::AMCL_unit`, [27](#)
 - `_y_base`
 - `fuse_G_A::AMCL_unit`, [23](#)
 - `select_A::AMCL_unit`, [27](#)
- `amcl_global_update`
 - `select_A`, [12](#)
- `amcl_initial_callback`
 - `select_A`, [13](#)
- `amcl_initial_update`
 - `select_A`, [13](#)
- `amcl_nomotion_update`
 - `select_A`, [13](#)
- `amcl_pose`
 - `fuse_G_A`, [10](#)
- `amcl_process`
 - `select_A`, [13](#)
- `amcl_recovery`
 - `select_A`, [14](#)
- `angle_diff`
 - `select_A`, [14](#)
- `choice`
 - `fuse_G_A::SELECTED`, [52](#)
 - `select_A::SELECTED`, [49](#)
- `crc_novetal`
 - `serial_af`, [19](#)
- `encoder_time`
 - `serial_af::Encoder`, [30](#)
- `fill`
 - `serial_af::Fuse_GPS_Odom`, [31](#)
- `fix`
 - `fuse_G_A::AMCL_unit`, [22](#)
 - `select_A::AMCL_unit`, [26](#)
- `flag_robot_move`
 - `select_A`, [16](#)
- `fuse_G_A.AMCL_unit`, [21](#)
- `fuse_G_A.GPS_unit`, [40](#)
- `fuse_G_A.Odom_unit`, [42](#)
- `fuse_G_A.py`, [55](#)
- `fuse_G_A.SELECTED`, [50](#)
- `fuse_G_A::AMCL_unit`
 - `__init__`, [22](#)
 - `_theta_base`, [23](#)
 - `_x_base`, [23](#)
 - `_y_base`, [23](#)
 - `fix`, [22](#)
 - `output`, [22](#)
 - `theta`, [24](#)
 - `theta_std`, [24](#)
 - `update`, [23](#)
 - `x`, [24](#)
 - `x_std`, [24](#)
 - `y`, [24](#)
 - `y_std`, [24](#)
- `fuse_G_A::GPS_unit`
 - `__init__`, [40](#)
 - `output`, [41](#)
 - `theta`, [41](#)
 - `theta_std`, [41](#)
 - `update`, [41](#)
 - `x`, [42](#)
 - `x_std`, [42](#)
 - `y`, [42](#)
 - `y_std`, [42](#)
- `fuse_G_A::Odom_unit`
 - `__init__`, [43](#)
 - `output`, [43](#)
 - `theta`, [44](#)
 - `theta_std`, [44](#)
 - `x`, [44](#)
 - `x_std`, [44](#)
 - `y`, [44](#)
 - `y_std`, [44](#)
- `fuse_G_A::SELECTED`
 - `__init__`, [51](#)
 - `choice`, [52](#)
 - `fuse_tf`, [52](#)
 - `output`, [51](#)
 - `set_map`, [52](#)

- set_odom, [52](#)
 - theta, [52](#)
 - theta_choice, [53](#)
 - x, [53](#)
 - x_choice, [53](#)
 - y, [53](#)
 - y_choice, [53](#)
- fuse_G_A, [7](#)
 - amcl_pose, [10](#)
 - gps_pose, [10](#)
 - listener, [10](#)
 - MAX_STD, [10](#)
 - make_choice, [8](#)
 - matrix_from_theta, [8](#)
 - normalization, [8](#)
 - odom_pose, [10](#)
 - out_pose, [11](#)
 - pub, [11](#)
 - sendTransform, [9](#)
 - t1, [11](#)
 - t2, [11](#)
 - tfBuffer, [11](#)
 - theta_from_matrix, [9](#)
 - tran_mat44, [9](#)
 - tran_theta_T, [10](#)
- fuse_encoder
 - serial_af::Fuse_GPS_Odom, [31](#)
- fuse_gps
 - serial_af::Fuse_GPS_Odom, [32](#)
- fuse_gps_odom
 - serial_af, [20](#)
- fuse_gps_odom_msg
 - serial_af::Fuse_GPS_Odom, [32](#)
- fuse_tf
 - fuse_G_A::SELECTED, [52](#)
 - select_A::SELECTED, [49](#)
- gps_pose
 - fuse_G_A, [10](#)
- gps_time
 - serial_af::GPS, [34](#)
- hdg_std
 - serial_af::GPS, [34](#)
- heading
 - serial_af::GPS, [34](#)
- hgt
 - serial_af::GPS, [34](#)
- hgt_std
 - serial_af::GPS, [35](#)
- hor_spd
 - serial_af::GPS, [35](#)
- lat
 - serial_af::GPS, [35](#)
- lat_std
 - serial_af::GPS, [35](#)
- left_encoder_val
 - serial_af::Encoder, [30](#)
- listener
 - fuse_G_A, [10](#)
- lock
 - select_A::AMCL_unit, [26](#)
 - select_A::GPS_unit, [38](#)
- lon
 - serial_af::GPS, [35](#)
- lon_std
 - serial_af::GPS, [35](#)
- MAX_STD
 - fuse_G_A, [10](#)
 - select_A, [17](#)
- main
 - select_A, [14](#)
 - serial_af, [18](#)
- make_choice
 - fuse_G_A, [8](#)
 - select_A, [14](#)
- matrix_from_theta
 - fuse_G_A, [8](#)
 - select_A, [14](#)
- mutex
 - serial_af, [20](#)
- normalization
 - fuse_G_A, [8](#)
- normalize
 - select_A, [15](#)
- odom_pose
 - fuse_G_A, [10](#)
- out_pose
 - fuse_G_A, [11](#)
- output
 - fuse_G_A::AMCL_unit, [22](#)
 - fuse_G_A::GPS_unit, [41](#)
 - fuse_G_A::Odom_unit, [43](#)
 - fuse_G_A::SELECTED, [51](#)
 - select_A::AMCL_unit, [26](#)
 - select_A::GPS_unit, [38](#)
 - select_A::Odom_unit, [46](#)
 - select_A::SELECTED, [48](#)
- plot
 - serial_af::Encoder, [30](#)
 - serial_af::GPS, [34](#)
- Pos_type
 - serial_af::GPS, [36](#)
- pub
 - fuse_G_A, [11](#)
- range_pub
 - serial_af, [18](#)
- right_encoder_val
 - serial_af::Encoder, [30](#)
- STATIC_LIMIT_ANGULAR
 - select_A, [17](#)
- STATIC_LIMIT_LINEAR

- select_A, 17
- SVs
 - serial_af::GPS, 36
- select_A.AMCL_unit, 25
- select_A.GPS_unit, 37
- select_A.Odom_unit, 45
- select_A.py, 56
- select_A.SELECTED, 47
- select_A::AMCL_unit
 - __init__, 26
 - _theta_base, 27
 - _x_base, 27
 - _y_base, 27
 - fix, 26
 - lock, 26
 - output, 26
 - theta, 28
 - theta_std, 28
 - update, 27
 - x, 28
 - x_std, 28
 - y, 28
 - y_std, 28
- select_A::GPS_unit
 - __init__, 38
 - lock, 38
 - output, 38
 - theta, 39
 - theta_std, 39
 - update, 38
 - x, 39
 - x_std, 39
 - y, 39
 - y_std, 39
- select_A::Odom_unit
 - __init__, 45
 - output, 46
 - theta, 46
 - theta_std, 46
 - x, 46
 - x_std, 46
 - y, 47
 - y_std, 47
- select_A::SELECTED
 - __init__, 48
 - choice, 49
 - fuse_tf, 49
 - output, 48
 - set_map, 48
 - set_odom, 49
 - theta, 49
 - theta_choice, 49
 - x, 50
 - x_choice, 50
 - y, 50
 - y_choice, 50
- select_A, 11
 - amcl_global_update, 12
 - amcl_initial_callback, 13
 - amcl_initial_update, 13
 - amcl_nomotion_update, 13
 - amcl_process, 13
 - amcl_recovery, 14
 - angle_diff, 14
 - flag_robot_move, 16
 - MAX_STD, 17
 - main, 14
 - make_choice, 14
 - matrix_from_theta, 14
 - normalize, 15
 - STATIC_LIMIT_ANGULAR, 17
 - STATIC_LIMIT_LINEAR, 17
 - sendTransform, 15
 - theta_from_matrix, 15
 - tran_mat44, 16
 - tran_theta_T, 16
- sendTransform
 - fuse_G_A, 9
 - select_A, 15
- serial_af, 17
 - crc_novetel, 19
 - fuse_gps_odom, 20
 - main, 18
 - mutex, 20
 - range_pub, 18
 - serial_process, 18
 - sound_range, 20
 - sound_range_init, 19
 - sound_seq, 20
- serial_af.Encoder, 29
- serial_af.Fuse_GPS_Odom, 30
- serial_af.GPS, 32
- serial_af.py, 57
- serial_af::Encoder
 - __init__, 29
 - encoder_time, 30
 - left_encoder_val, 30
 - plot, 30
 - right_encoder_val, 30
- serial_af::Fuse_GPS_Odom
 - __init__, 31
 - fill, 31
 - fuse_encoder, 31
 - fuse_gps, 32
 - fuse_gps_odom_msg, 32
- serial_af::GPS
 - __init__, 33
 - gps_time, 34
 - hdg_std, 34
 - heading, 34
 - hgt, 34
 - hgt_std, 35
 - hor_spd, 35
 - lat, 35
 - lat_std, 35
 - lon, 35

- lon_std, 35
- plot, 34
- Pos_type, 36
- SVs, 36
- Slo_stat, 36
- solnSVs, 36
- Trk_gnd, 36
- Vert_spd, 36
- serial_process
 - serial_af, 18
- set_map
 - fuse_G_A::SELECTED, 52
 - select_A::SELECTED, 48
- set_odom
 - fuse_G_A::SELECTED, 52
 - select_A::SELECTED, 49
- Slo_stat
 - serial_af::GPS, 36
- solnSVs
 - serial_af::GPS, 36
- sound_range
 - serial_af, 20
- sound_range_init
 - serial_af, 19
- sound_seq
 - serial_af, 20
- t1
 - fuse_G_A, 11
- t2
 - fuse_G_A, 11
- tfBuffer
 - fuse_G_A, 11
- theta
 - fuse_G_A::AMCL_unit, 24
 - fuse_G_A::GPS_unit, 41
 - fuse_G_A::Odom_unit, 44
 - fuse_G_A::SELECTED, 52
 - select_A::AMCL_unit, 28
 - select_A::GPS_unit, 39
 - select_A::Odom_unit, 46
 - select_A::SELECTED, 49
- theta_choice
 - fuse_G_A::SELECTED, 53
 - select_A::SELECTED, 49
- theta_from_matrix
 - fuse_G_A, 9
 - select_A, 15
- theta_std
 - fuse_G_A::AMCL_unit, 24
 - fuse_G_A::GPS_unit, 41
 - fuse_G_A::Odom_unit, 44
 - select_A::AMCL_unit, 28
 - select_A::GPS_unit, 39
 - select_A::Odom_unit, 46
- tran_mat44
 - fuse_G_A, 9
 - select_A, 16
- tran_theta_T
 - fuse_G_A, 10
 - select_A, 16
- Trk_gnd
 - serial_af::GPS, 36
- update
 - fuse_G_A::AMCL_unit, 23
 - fuse_G_A::GPS_unit, 41
 - select_A::AMCL_unit, 27
 - select_A::GPS_unit, 38
- Vert_spd
 - serial_af::GPS, 36
- x
 - fuse_G_A::AMCL_unit, 24
 - fuse_G_A::GPS_unit, 42
 - fuse_G_A::Odom_unit, 44
 - fuse_G_A::SELECTED, 53
 - select_A::AMCL_unit, 28
 - select_A::GPS_unit, 39
 - select_A::Odom_unit, 46
 - select_A::SELECTED, 50
- x_choice
 - fuse_G_A::SELECTED, 53
 - select_A::SELECTED, 50
- x_std
 - fuse_G_A::AMCL_unit, 24
 - fuse_G_A::GPS_unit, 42
 - fuse_G_A::Odom_unit, 44
 - select_A::AMCL_unit, 28
 - select_A::GPS_unit, 39
 - select_A::Odom_unit, 46
- y
 - fuse_G_A::AMCL_unit, 24
 - fuse_G_A::GPS_unit, 42
 - fuse_G_A::Odom_unit, 44
 - fuse_G_A::SELECTED, 53
 - select_A::AMCL_unit, 28
 - select_A::GPS_unit, 39
 - select_A::Odom_unit, 47
 - select_A::SELECTED, 50
- y_choice
 - fuse_G_A::SELECTED, 53
 - select_A::SELECTED, 50
- y_std
 - fuse_G_A::AMCL_unit, 24
 - fuse_G_A::GPS_unit, 42
 - fuse_G_A::Odom_unit, 44
 - select_A::AMCL_unit, 28
 - select_A::GPS_unit, 39
 - select_A::Odom_unit, 47