模式: 全部

左边文件: G:\ros1111111\aoa_ros\src\aoa_ros.cpp 右边文件: G:\aoa_ros\src\aoa_ros.cpp

1	#include "aoa_ros.h"	=	1	#include "aoa_ros.h"
2			2	
3			3	
4	bool AOA_ros::WriteToUart(unsigned cha » r*)		4	bool AOA_ros::WriteToUart(unsigned cha » r*)
5	{		5	{
6	return true;		6	return true;
7	}		7	}
8	AOA_ros::AOA_ros()		8	AOA_ros::AOA_ros()
9	{		9	{
10			10	
11	/* Get Luncher file define value * » /		11	/* Get Luncher file define value * » /
12	ros::NodeHandle nh_private("~");	<>	12	m_hNodeAOA = Node::make_shared("ao » a_ros");
13			13	
			14	m_strUsart_ports = "/dev/aoa_ros";
14	nh_private.param<std::string>("ser » ial_port", m_strUsart_ports, "/dev/a » oa_ros");		15	m_hNodeAOA->get_parameter("serial_ » port", m_strUsart_ports);
			16	m_nBaud_rate = 115200;
15	nh_private.param<int>("serial_baud » rate", m_nBaud_rate, 115200);		17	m_hNodeAOA->get_parameter("serial_ » baudrate", m_nBaud_rate);
16			18	
17		=	19	
18	m_pAOA_pub = m_hNodeAOA.advertise< » std_msgs::Int32MultiArray>("/AOA_rep » ort_date", 1000);	<>	20	m_pAOA_pub = m_hNodeAOA->create_pu » blisher<std_msgs::msg::Int32MultiArr » ay>("AOA_report_date", 7);
			21	
19	}	=	22	}
20	bool AOA_ros::OpenSerial(void)		23	bool AOA_ros::OpenSerial(void)
21	{		24	{
22	try		25	try
23	{		26	{
24	m_Robot_Serial.setPort(m_strUs » art_ports);		27	m_Robot_Serial.setPort(m_strUs » art_ports);
25	m_Robot_Serial.setBaudrate(m_n » Baud_rate);		28	m_Robot_Serial.setBaudrate(m_n » Baud_rate);
26	serial::Timeout to = serial::T » imeout::simpleTimeout(2000);		29	serial::Timeout to = serial::T » imeout::simpleTimeout(2000);
27	m_Robot_Serial.setTimeout(to);		30	m_Robot_Serial.setTimeout(to);
28	m_Robot_Serial.open();		31	m_Robot_Serial.open();
29	}		32	}
30	catch (serial::IOException& e)		33	catch (serial::IOException& e)
31	{		34	{
		<>	35	
32	ROS_ERROR_STREAM("[OpenSerial] » Unable to open port ");		36	RCLCPP_ERROR(m_hNodeAOA->get_l » ogger(),"[OpenSerial] Unable to open » port ");
33	return false;	=	37	return false;
34	}		38	}
35			39	
36	if(m_Robot_Serial.isOpen())		40	if(m_Robot_Serial.isOpen())
37	{		41	{
38	ROS_INFO_STREAM("[OpenSerial] » Serial Port opened");	<>	42	RCLCPP_INFO(m_hNodeAOA->get_lo » gger() ,"[OpenSerial] Serial Port op » ened");

左边文件: G:\ros1111111\aoa_ros\src\aoa_ros.cpp 右边文件: G:\aoa_ros\src\aoa_ros.cpp (继续)

39	`        return true;`	=	43	`        return true;`
40	`    }`		44	`    }`
41	`    else`		45	`    else`
42	`    {`		46	`    {`
43	`        ROS_INFO_STREAM("[OpenSerial] Serial Port open fail");`	<>	47	`        RCLCPP_ERROR(m_hNodeAOA->get_logger() ,"[OpenSerial] Serial Port open fail");`
44	`        return false;`	=	48	`        return false;`
45	`    }`		49	`    }`
46	`}`		50	`}`
47	`bool AOA_ros::ReadFromUart(void)`		51	`bool AOA_ros::ReadFromUart(void)`
48	`{`		52	`{`
49	`    AOA_Serial_Data_Union Reciver_data ;`		53	`    AOA_Serial_Data_Union Reciver_data ;`
50	`    //Reciver_data.clear();`		54	`    //Reciver_data.clear();`
51	`    memset(&Reciver_data,0,sizeof(AOA_Serial_Data_Union));`		55	`    memset(&Reciver_data,0,sizeof(AOA_Serial_Data_Union));`
52	`    unsigned char RosReadSerialBuffer[1];`		56	`    unsigned char RosReadSerialBuffer[1];`
53	`    std_msgs::Int32MultiArray  AOA_msg ;`	<>	57	`    std_msgs::msg::Int32MultiArray  AOA_msg;`
54	`    float angle_f = 0.0;`	=	58	`    float angle_f = 0.0;`
55	`    int angle_n = 0;`		59	`    int angle_n = 0;`
56			60	
57			61	
58	`    if(m_Robot_Serial.available())`		62	`    if(m_Robot_Serial.available())`
59	`    {`		63	`    {`
60	`        //ROS_INFO_STREAM("Reading from serial port\n");`		64	`        //ROS_INFO_STREAM("Reading from serial port\n");`
61	`        m_Robot_Serial.read(Reciver_data.buffer,sizeof(Reciver_data.buffer));`		65	`        m_Robot_Serial.read(Reciver_data.buffer,sizeof(Reciver_data.buffer));`
62	`        int start = Reciver_data.AOA_report_date.title.start;`	+-		
63		=	66	
64			67	
65	`        if (Reciver_data.AOA_report_date.title.start == 0x59`		68	`        if (Reciver_data.AOA_report_date.title.start == 0x59`
66	`        && Reciver_data.AOA_report_date.title.len == 0x13`		69	`        && Reciver_data.AOA_report_date.title.len == 0x13`
67	`        && Reciver_data.AOA_report_date.type == 0x63`		70	`        && Reciver_data.AOA_report_date.type == 0x63`
68	`        && Reciver_data.AOA_report_date.end.end == 0x47`		71	`        && Reciver_data.AOA_report_date.end.end == 0x47`
69	`        )`		72	`        )`
70	`        {`		73	`        {`
71	`            //check key`		74	`            //check key`
72	`            AOA_msg.data.clear();`		75	`            AOA_msg.data.clear();`
73	`            AOA_msg.data.push_back(Reciver_data.AOA_report_date.rx_rssi_first);`		76	`            AOA_msg.data.push_back(Reciver_data.AOA_report_date.rx_rssi_first);`
74	`            AOA_msg.data.push_back(Reciver_data.AOA_report_date.rx_rssi_all);`		77	`            AOA_msg.data.push_back(Reciver_data.AOA_report_date.rx_rssi_all);`
75	`            AOA_msg.data.push_back(Reciver_data.AOA_report_date.battery);`		78	`            AOA_msg.data.push_back(Reciver_data.AOA_report_date.battery);`
76	`            AOA_msg.data.push_back(Reciver_data.AOA_report_date.keys);`		79	`            AOA_msg.data.push_back(Reciver_data.AOA_report_date.keys);`

左边文件: G:\ros1111111\aoa_ros\src\aoa_ros.cpp 右边文件: G:\aoa_ros\src\aoa_ros.cpp (继续)

77	AOA_msg.data.push_back(Rec » iver_data.AOA_report_date.dist);		80	AOA_msg.data.push_back(Rec » iver_data.AOA_report_date.dist);
78	angle_f = Reciver_data.AOA » _report_date.angle /1000.0;	<>	81	angle_f = Reciver_data.AOA » _report_date.angle /1000.0; // NOLIN » T(bugprone-narrowing-conversions)
79	angle_n = angle_f*180/3.14 » ;	=	82	angle_n = angle_f*180/3.14 » ;
80	AOA_msg.data.push_back(ang » le_n);		83	AOA_msg.data.push_back(ang » le_n);
81	AOA_msg.data.push_back(Rec » iver_data.AOA_report_date.anchor_sta » tus);		84	AOA_msg.data.push_back(Rec » iver_data.AOA_report_date.anchor_sta » tus);
82	AOA_msg.data.push_back(Rec » iver_data.AOA_report_date.quality);		85	AOA_msg.data.push_back(Rec » iver_data.AOA_report_date.quality);
83			86	
84	m_pAOA_pub.publish(AOA_msg »);	<>	87	m_pAOA_pub->publish(AOA_ms » g);
85		=	88	
86	}		89	}
87	else		90	else
88	{		91	{
89	m_Robot_Serial.read(RosRea » dSerialBuffer,sizeof(RosReadSerialBu » ffer));		92	m_Robot_Serial.read(RosRea » dSerialBuffer,sizeof(RosReadSerialBu » ffer));
90	ROS_INFO_STREAM("[ReadFrom » Uart] data is illegle !!!\n");	<>	93	RCLCPP_ERROR(m_hNodeAOA->g » et_logger() ,"[ReadFromUart] data is » illegle !!!\n");
91	}	=	94	}
92	}		95	}
93	else		96	else
94	{		97	{
95	//ROS_INFO_STREAM("serial is u » navailable !!!\n");		98	//ROS_INFO_STREAM("serial is u » navailable !!!\n");
96	}		99	}
97			100	
98	return false;		101	return false;
99	}		102	}
100	bool AOA_ros::LoopProcess(void)		103	bool AOA_ros::LoopProcess(void)
101	{		104	{
102			105	
103	ros::Rate loop_rate(100);	<>	106	rclcpp::Rate loop_rate(100);
104		=	107	
105	while (ros::ok())	<>	108	while (rclcpp::ok())
106	{	=	109	{
107	//main logic		110	//main logic
108			111	
109	//read urat		112	//read urat
110	ReadFromUart();		113	ReadFromUart();
111			114	
112	ros::spinOnce();	<>	115	rclcpp::spin_some(m_hNodeAOA);
113	loop_rate.sleep();	=	116	loop_rate.sleep();
114	}		117	}
115			118	
116	return true;		119	return true;
117	}		120	}
118	AOA_ros::~AOA_ros()		121	AOA_ros::~AOA_ros()
119	{		122	{
120	//do nothing		123	//do nothing

左边文件: G:\ros1111111\aoa_ros\src\aoa_ros.cpp 右边文件: G:\aoa_ros\src\aoa_ros.cpp (继续)

121	}		124	}
122	int main(int argc, char *argv[])		125	int main(int argc, char *argv[])
123	{		126	{
124	//init		127	//init
125	ros::init(argc, argv, "AOA_ros");	<>	128	rclcpp::init(argc, argv);
126		=	129	
127	//creater		130	//creater
128	AOA_ros AOA_ros_control;		131	AOA_ros AOA_ros_control;
129			132	
130	//open serial		133	//open serial
131	if (!AOA_ros_control.OpenSerial())		134	if (!AOA_ros_control.OpenSerial())
132	{		135	{
133	return false;	<>	136	return 0;
134	}	=	137	}
135			138	
136	//go to main loop		139	//go to main loop
137	AOA_ros_control.LoopProcess();		140	AOA_ros_control.LoopProcess();
138			141	
139			142	
140			143	
141	return 0;		144	return 0;
142			145	
143	}		146	}
144			147	
145			148	
146			149	
147			150	
148			151	
149			152	