AAE3004
Dynamical Systems and Control

Lab 1b: Autonomous ROS car control project

Instructors: Dr Bing XU, Mr Jian LIU (TA), and Mr Sergio VICENZO (TA)
Research Assistant Professor
Department of Aeronautical and Aviation Engineering
The Hong Kong Polytechnic University
Office: QR832
Tel: 3400 8010
Email: pbing.xu@polyu.edu.hk

AAE3004 Lab 1 Project Report Requirement

Assessment rubrics	Mark range[Note 1]	Outcomes
Excellent	[8,10]	• Be able to successfully apply control theory (e.g. PID) to improve ROS car performance[Note 2] • Be able to present ROS car project learning experience appropriately, including but not limited to new skills/knowledge, troubleshooting, etc.
Good	[6-8)	• Be able to apply control theory (e.g. PID) to improve ROS car performance[Note 3] • Be able to present ROS car project learning experience, including but not limited to new skills/knowledge, troubleshooting, etc.
Satisfactory	[4-6)	• Be able to present ROS car project learning experience, including but not limited to new skills/knowledge, troubleshooting, etc.
Unsatisfactory	[0,4]	• Not able to apply control theory to ROS car, nor present ROS car project learning experience, including but not limited to new skills/knowledge, troubleshooting, etc.

Note 1: Full mark: 10

Note 2: Open problem, No standard solution.

Note 3: Trials might be unsuccessful.

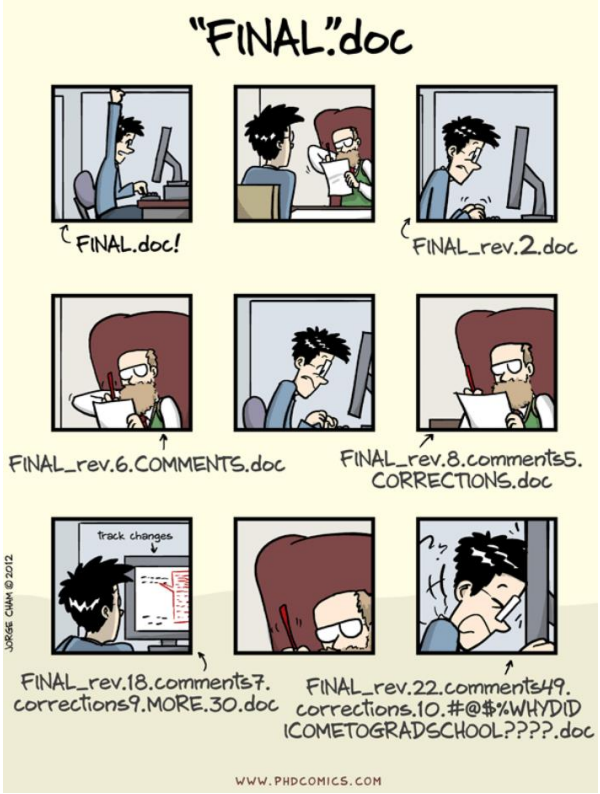
AAE3004

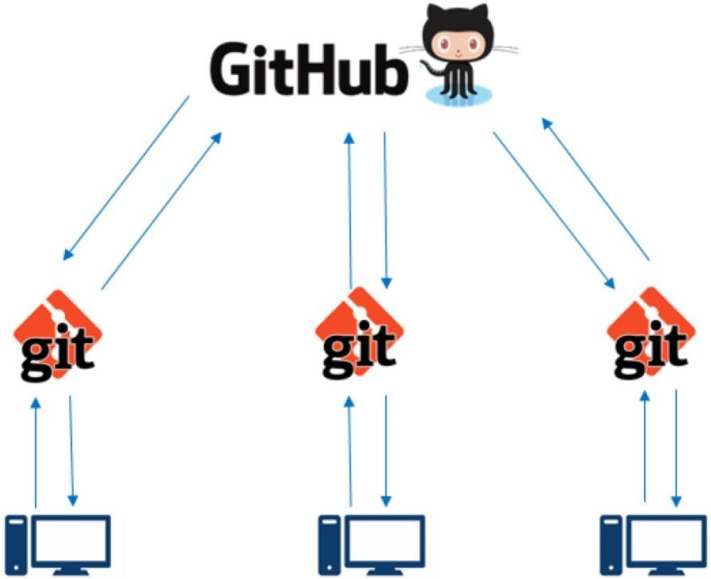
- How to use github
- Code Structure——ROS
- Code Structure——STM32

AAE3004

How to use github ?

Has this happened to you?

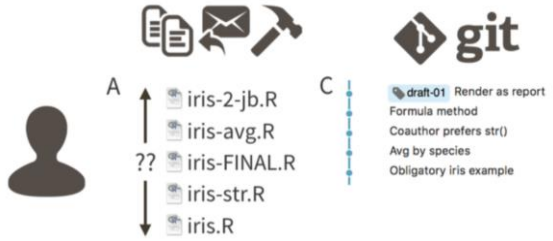




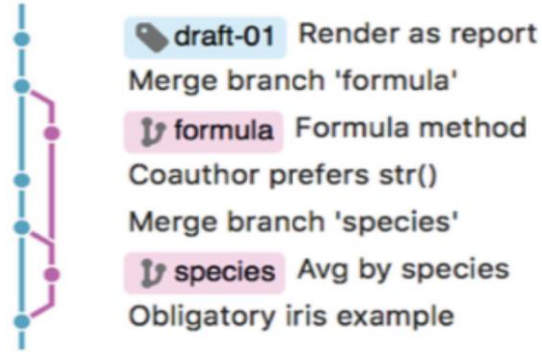
Git and **GitHub** allow for easy management and sharing of data analytic content.

What Git can do ?

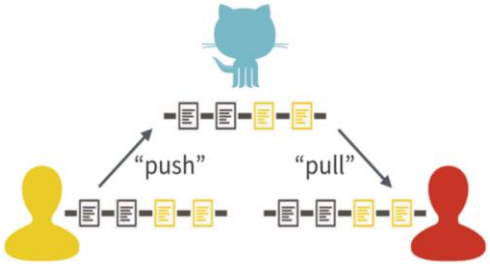
Git allows for tracking of a project through **commit** messages



You can use **branches** when working as a team or on a new feature

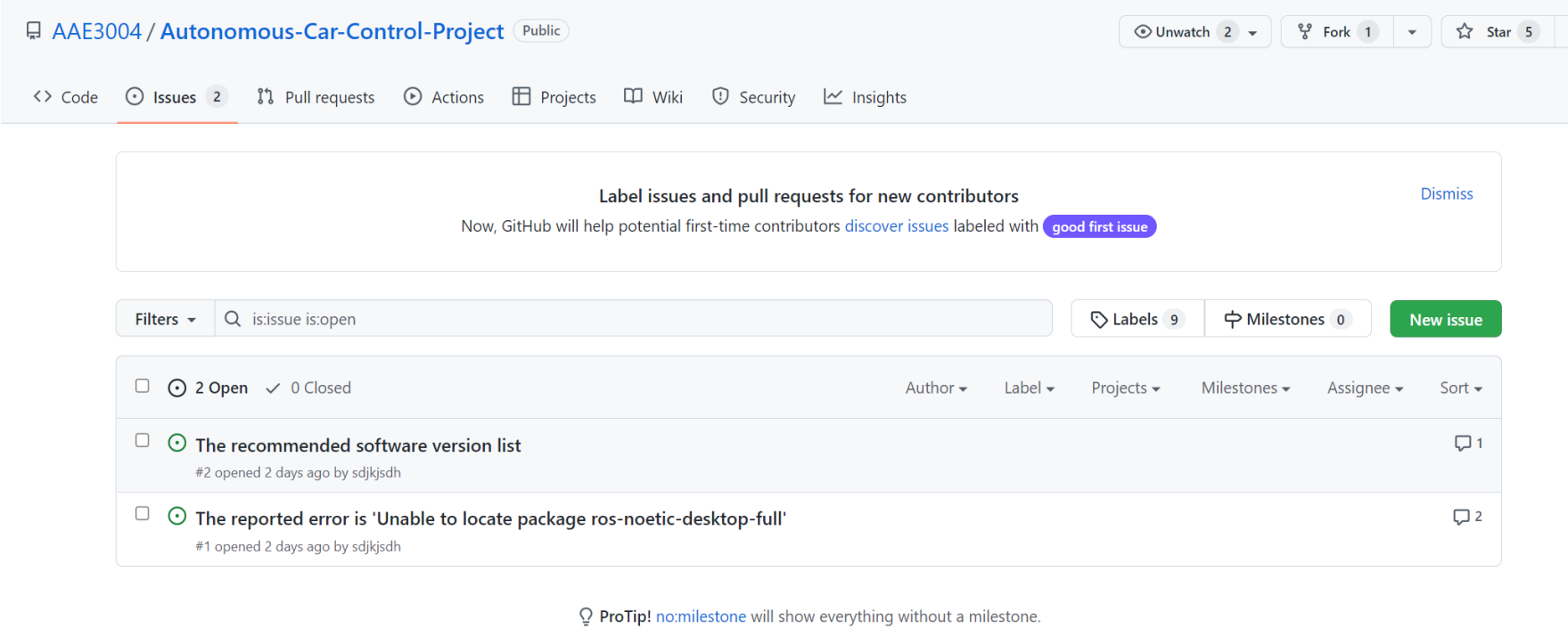


GitHub allows for easy **collaboration** on projects



What Git can do ?

You can also submit issues if you're having trouble with software

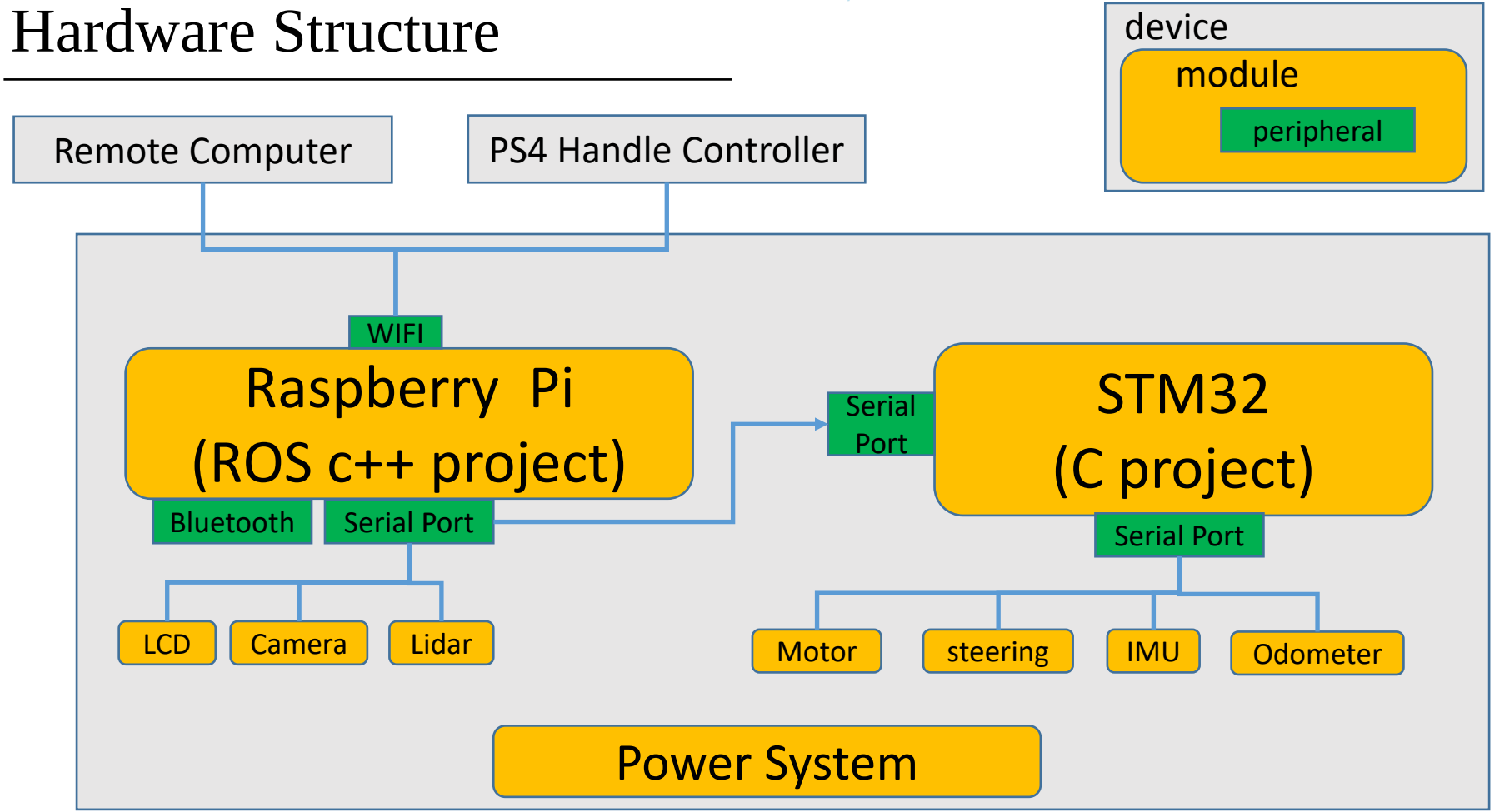


References

- https://happygitwithr.com/
  - Extremely useful tool for learning Git and GitHub—based off using R/Rstudio for analyses
- https://learngitbranching.js.org/
  - Can help you visualize the process of using Git
- "Excuse Me, Do You Have a Moment to Talk About Version Control?" (Bryan 2017)
  - Great paper for telling you why you should use Git and GitHub

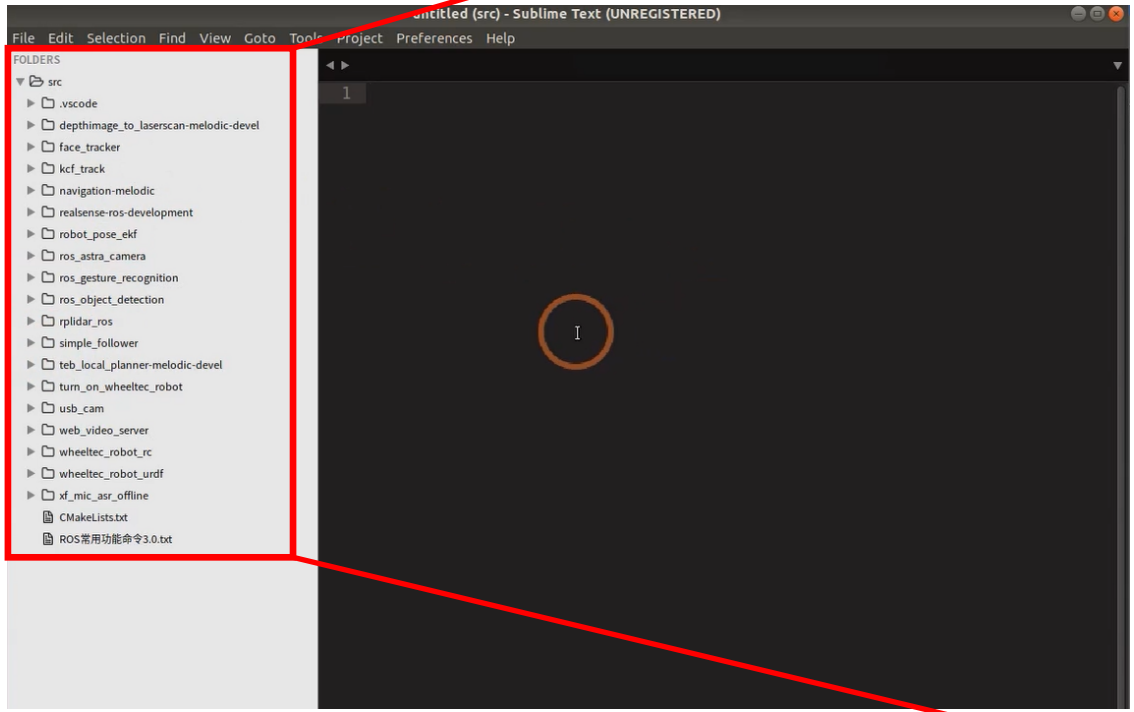
Code Structure—ROS

Hardware Structure

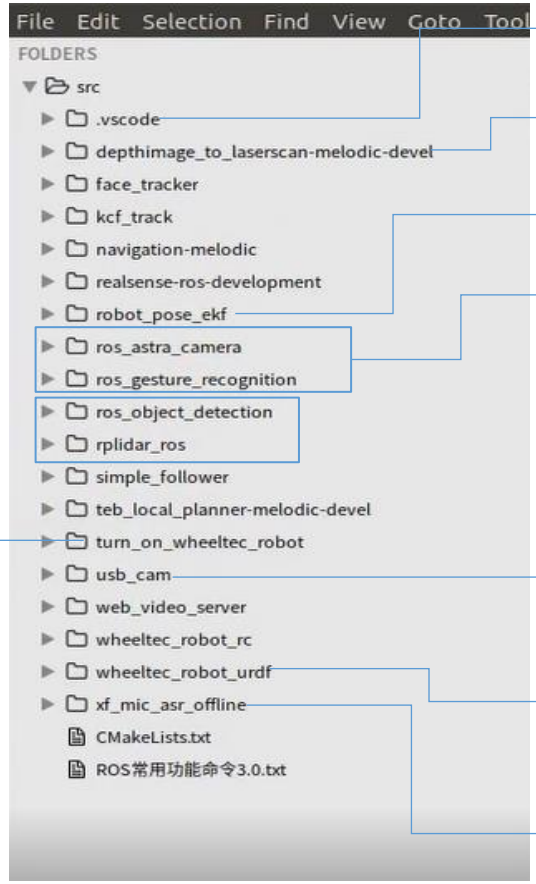


software structure of ROS

- Suppose you have completed the steps in Section 6.3 and 6.4 of the 'User Guide'



software structure of ROS



Vscode Project Envrionment

Laser Package

pose estimation Package

Camera Package

Enable and Initialize ROS on the car

XML format for representing robot model

Interacting with hardware

how to check specific code

A demo: Check real-time information of odometer by ROS topic

- Login in by SSH command
- Start the initialization node:

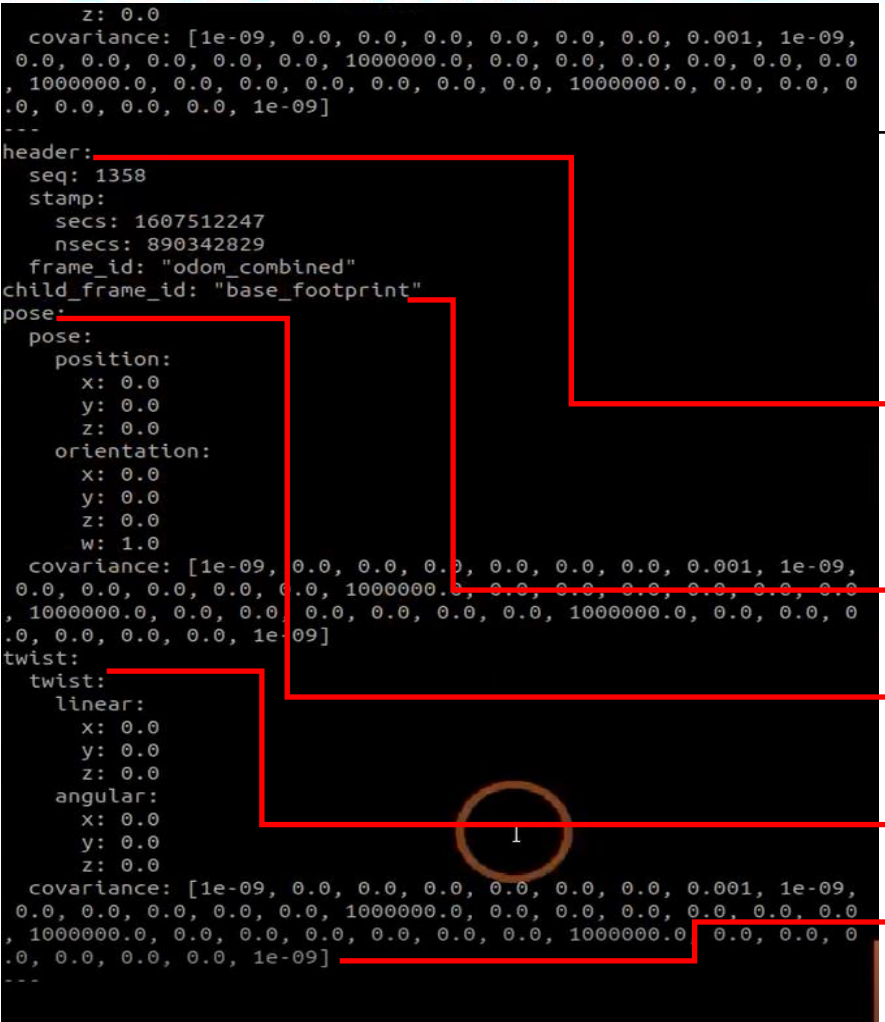
  $ roslaunch turn_on_wheeltec_robot turn_on_wheeltec_robot.launch

- Check the existing topics:

  $ rostopic list

```
passoni@passoni:~$ rostopic list
/PowerVoltage
/amcl_pose
/cmd_vel
/joint_states
/mobile_base/sensors/imu_data
/odom
/pose
/robot_pose_ekf/odom_combined
/rosout
/rosout_agg
/tf
/tf_static
passoni@passoni:~$
```

/odom: a ros topic for odometer

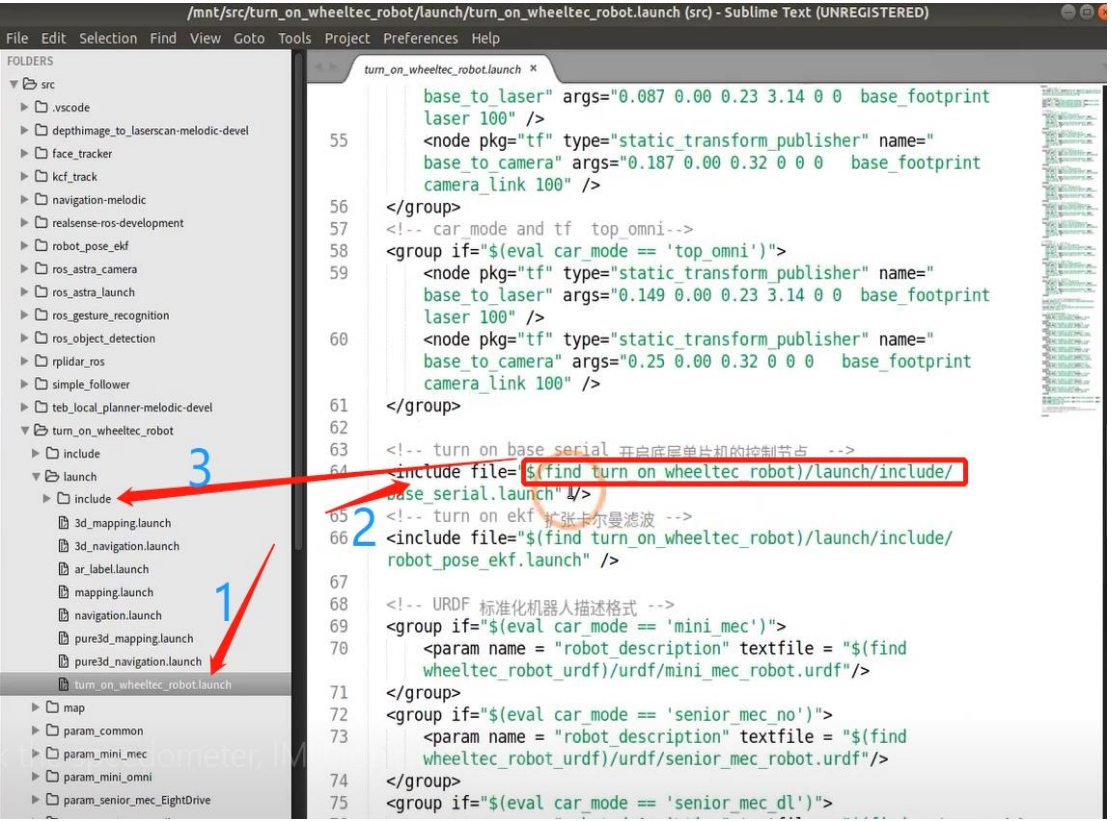


check the odometer information:

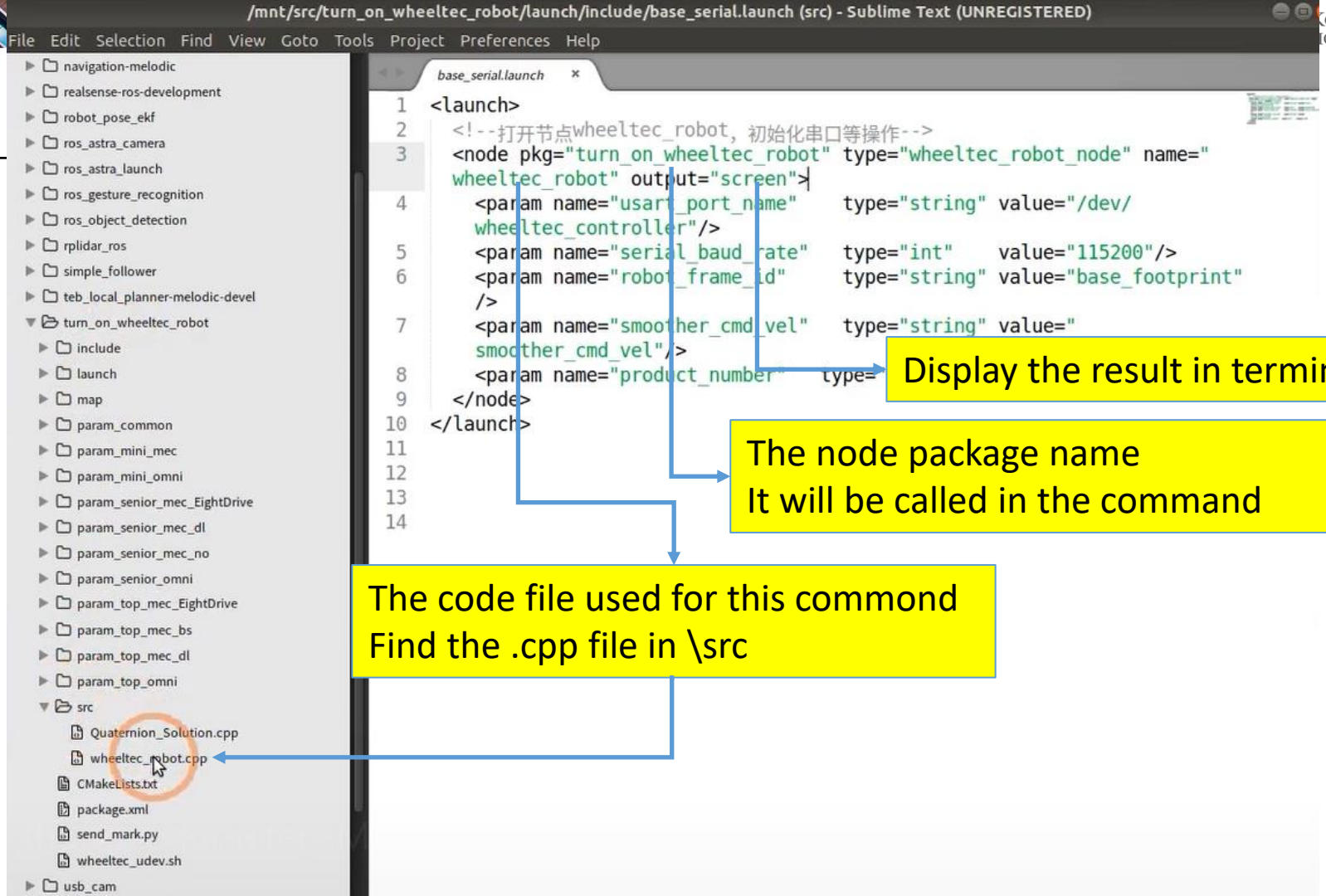
$ rostopic echo /odom

Information Structure:

- **Header**: sequence number and timestamp
- **child_frame_id**: local coordinate id
- **pose**: 3D odometer move distance
- **twist**: 3D linear/angular velocity
- **Covariance**: uncertanty



- Remember the topic we started is turn_on_wheeltec_robot.launch
- find .launch file in the ROS function package
- Redirected to **the base_serial.launch**


/mnt/src/turn_on_wheeltec_robot/launch/include/base_serial.launch (src) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
navigation-melodic
realsense-ros-development
robot_pose_ekf
ros_astra_camera
ros_astra_launch
ros_gesture_recognition
ros_object_detection
rplidar_ros
simple_follower
teb_local_planner-melodic-devel
turn_on_wheeltec_robot
include
launch
map
param_common
param_mini_mec
param_mini_omni
param_senior_mec_EightDrive
param_senior_mec_dl
param_senior_mec_no
param_senior_omni
param_top_mec_EightDrive
param_top_mec_bs
param_top_mec_dl
param_top_omni
src
Quaternion_Solution.cpp
wheeltec_robot.cpp
CMakeLists.txt
package.xml
send_mark.py
wheeltec_udev.sh
usb_cam
base_serial.launch
1 <launch>
2 <!--打开节点wheeltec_robot，初始化串口等操作-->
3 <node pkg="turn_on_wheeltec_robot" type="wheeltec_robot_node" name="
wheeltec_robot" output="screen">
4 <param name="usart_port_name" type="string" value="/dev/
wheeltec_controller"/>
5 <param name="serial_baud_rate" type="int" value="115200"/>
6 <param name="robot_frame_id" type="string" value="base_footprint"
/>
7 <param name="smoother_cmd_vel" type="string" value="
smoother_cmd_vel"/>
8 <param name="product_number" type=
9 </node>
10 </launch>
11
12
13
14
Display the result in terminal
The node package name
It will be called in the command
The code file used for this commond
Find the .cpp file in \src

/mnt/src/turn_on_wheeltec_robot/src/wheeltec_robot.cpp (src) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

navigation-melodic
realsense-ros-development
robot_pose_ekf
ros_astra_camera
ros_astra_launch
ros_gesture_recognition
ros_object_detection
rplidar_ros
simple_follower
teb_local_planner-melodic-devel
turn_on_wheeltec_robot
include
launch
map
param_common
param_mini_mec
param_mini_omni
param_senior_mec_EightDrive
param_senior_mec_dl
param_senior_mec_no
param_senior_omni
param_top_mec_EightDrive
param_top_mec_bs
param_top_mec_dl
param_top_omni
src
Quaternion_Solution.cpp
wheeltec_robot.cpp
CMakeLists.txt
package.xml
send_mark.py
wheeltec_udev.sh
usb_cam

wheeltec_robot.cpp

```
289   _Last_Time = ros::Time::now();
290   while(ros::ok())
291   {
292     _Now = ros::Time::now();
293     Sampling_Time = (_Now - _Last_Time).toSec();
294    //Sampling_time是采样时间，虽然下位机发送的数据频率是固定的，这里计算里程增量以
295     if (true == Get_Sensor_Data())  //从串口读取下位机法过来的全部数据
296     {
297       Robot_Pos.X+=(Robot_Vel.X * cos(Robot_Pos.Z) - Robot_Vel.Y * sin(R
298       Robot_Pos.Y+=(Robot_Vel.X * sin(Robot_Pos.Z) + Robot_Vel.Y * cos(R
299       Robot_Pos.Z+=Robot_Vel.Z * Sampling_Time; //角位移
300       Quaternion_Solution(Mpu6050.angular_velocity.x, Mpu6050.angular_ve
301                           Mpu6050.linear_acceleration.x, Mpu6050.linear_acceleratio
302       Publish_Odom();      //发布里程计话题
303       Publish_Pose();
304       Publish_ImuSensor();//发布话题
305       Publish_Voltage();//发布电源电压
306     }
307     _Last_Time = _Now;//记录时间
308     ros::spinOnce();//循环等待回调函数
309     }
310 }
311 /*************************************
312 Date: May 31, 2020
313 Function: 构造函数，只执行一次，用于初始
314 *************************************
315 turn_on_robot::turn_on_robot():Sampli
316 {
317   memset(&Robot_Pos, 0, sizeof(Robot_Pos));
318   memset(&Robot_Vel, 0, sizeof(Robot_Vel));
319   memset(&Receive_Data, 0, sizeof(Receive_Data)); //构造函数初始化
320   memset(&Send_Data, 0, sizeof(Send_Data));
```

- Publish_Odom(): odometer information
- Publish_Pose(): posture information
- Publish_ImuSensor(): imu information
- Publish_Voltage(): power system information

/mnt/src/turn_on_wheeltec_robot/src/wheeltec_robot.cpp (src) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

- navigation-melodic
- realsense-ros-development
- robot_pose_ekf
- ros_astra_camera
- ros_astra_launch
- ros_gesture_recognition
- ros_object_detection
- rplidar_ros
- simple_follower
- teb_local_planner-melodic-devel
- turn_on_wheeltec_robot
  - include
  - launch
  - map
  - param_common
  - param_mini_mec
  - param_mini_omni
  - param_senior_mec_EightDrive
  - param_senior_mec_dl
  - param_senior_mec_no
  - param_senior_omni
  - param_top_mec_EightDrive
  - param_top_mec_bs
  - param_top_mec_dl
  - param_top_omni
  - src
    - Quaternion_Solution.cpp
    - wheeltec_robot.cpp
  - CMakeLists.txt
  - package.xml
  - send_mark.py
  - wheeltec_udev.sh
- usb_cam
- web_video_server

wheeltec_robot.cpp

```
116 /**************************************
117 Date: May 31, 2020
118 Function: 发布里程计相关信息
119 **************************************/
120 void turn_on_robot::Publish_Odom()
121 {
122     geometry_msgs::Quaternion odom_quat = tf::createQuaternionMsgFromYaw
123     nav_msgs::Odometry odom;//里程计话题消息数据类型
124     odom.header.stamp = ros::Time::now();//当前时间
125     odom.header.frame_id = "odom_combined";
126     odom.pose.pose.position.x = Robot_Pos.X;//位置
127     odom.pose.pose.position.y = Robot_Pos.Y;
128     //odom.pose.pose.position.z = 0;
129     odom.pose.pose.position.z = Robot_Pos.Z;
130     odom.pose.pose.orientation = odom_quat;
131     //设置速度
132     odom.child_frame_id = robot_frame_id;
133     odom.twist.twist.linear.x =  Robot_Vel.X;//X方向前进速度
134     odom.twist.twist.linear.y =  Robot_Vel.Y;//y方向前进速度
135     odom.twist.twist.angular.z = Robot_Vel.Z;  //角速度
136     //这个矩阵有两种，分机器人静止和动起来的时候用 这是扩展卡尔曼滤波的，官网提供的2
137     if(Robot_Vel.X== 0&&Robot_Vel.Z== 0)//如果velocity是零，说明编码器的误差
138     memcpy(&odom.pose.covariance, odom_pose_covariance2, sizeof(odom_pos
139     memcpy(&odom.twist.covariance, odom_twist_covariance2, sizeof(odom_t
140     else//如果小车velocity非零，考虑到运动中编码器可能带来的滑动误差，认为imu的数
141     memcpy(&odom.pose.covariance, odom_pose_covariance, sizeof(odom_pose
142     memcpy(&odom.twist.covariance, odom_twist_covariance, sizeof(odom_tw
143     odom_publisher.publish(odom);//发布这个话题 消息类型是nav_msgs::Odometry
144 }
145
146 void turn_on_robot::Publish_Pose()
147 {
148     geometry_msgs::Quaternion odom_quat = tf::createQuaternionMsgFromYaw
```

Line 120, Column 21 Spaces: 2 C++

Open the function to check the cpp code

Code Structure—STM32

Install the uVision IDE follow the user guide on github

User manual for uVision installation

AAE3004 / **Autonomous-Car-Control-Project** Public

Code | Issues 2 | Pull requests | Actions | Projects | Wiki | Security | Insights

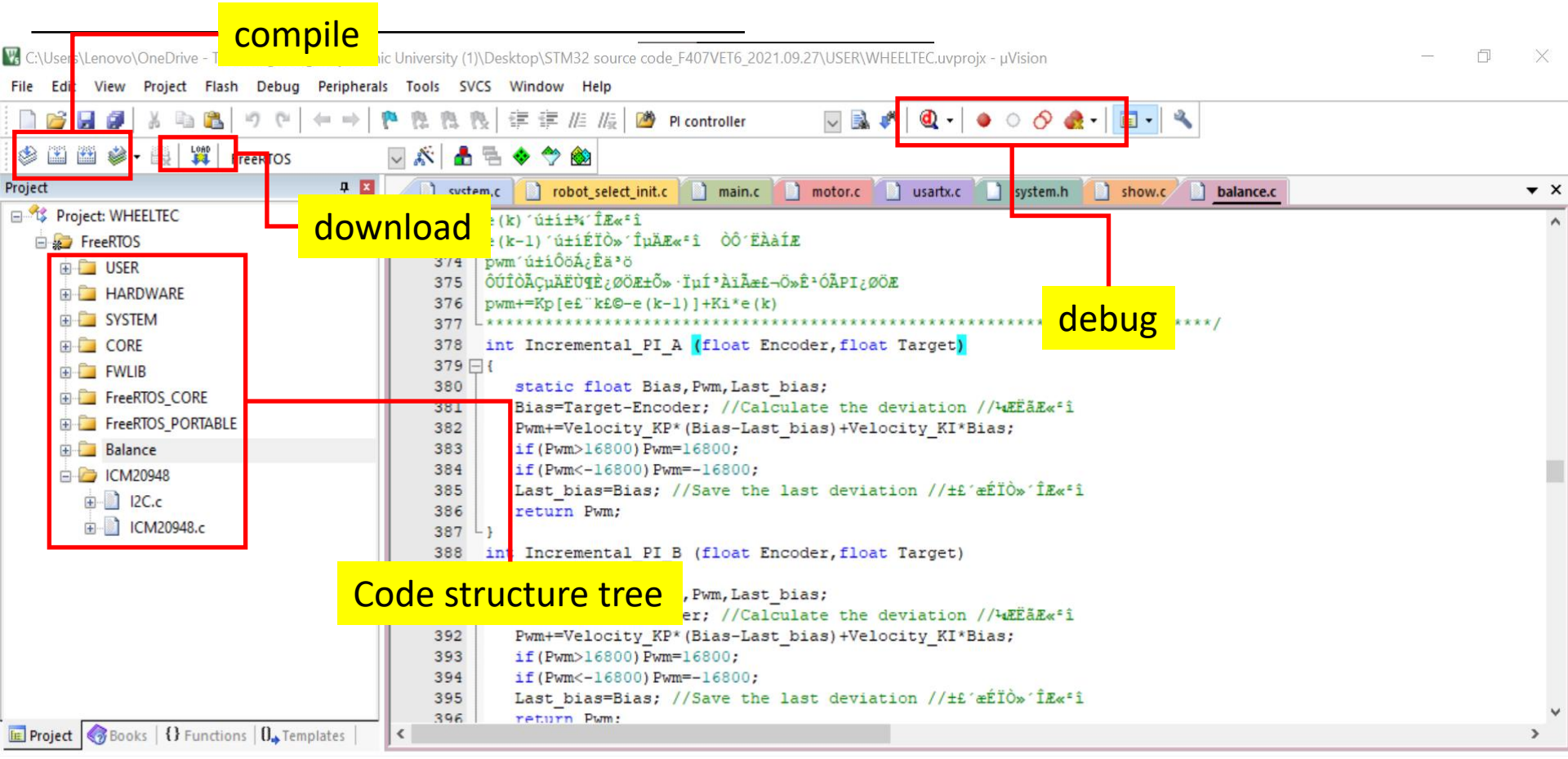
main ▾ Autonomous-Car-Control-Project / STM32 /

Go to file | Add file ▾ | ···

AAE3004 Add files via upload — 617aaf1 26 days ago — History

1. Download the program via USB	Add files via upload	27 days ago
AAE3004 - MANUAL FOR ACCESING STM32 SOURCE CO...	Add files via upload	27 days ago
AAE3004 - PID controller demonstration.mp4	Add files via upload	27 days ago
CP2102Driver.zip	Add files via upload	27 days ago
FlyMcu.exe	Add files via upload	26 days ago
STM32 source code_F407VET6_2021.09.27.zip	Add files via upload	27 days ago
User Guide.pdf	Add files via upload	27 days ago

The STM32 project is used to interacting with hardware and witten by c



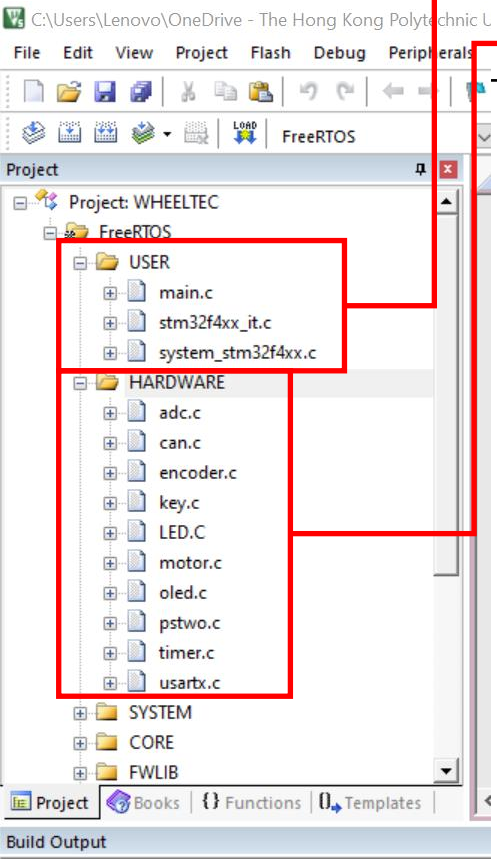
project Environment Initializaiton

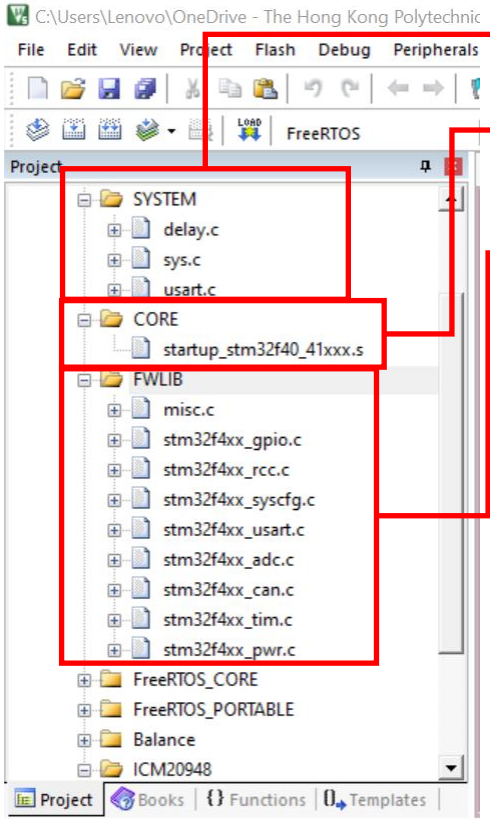
Peripheral configuration and function

System time and communication

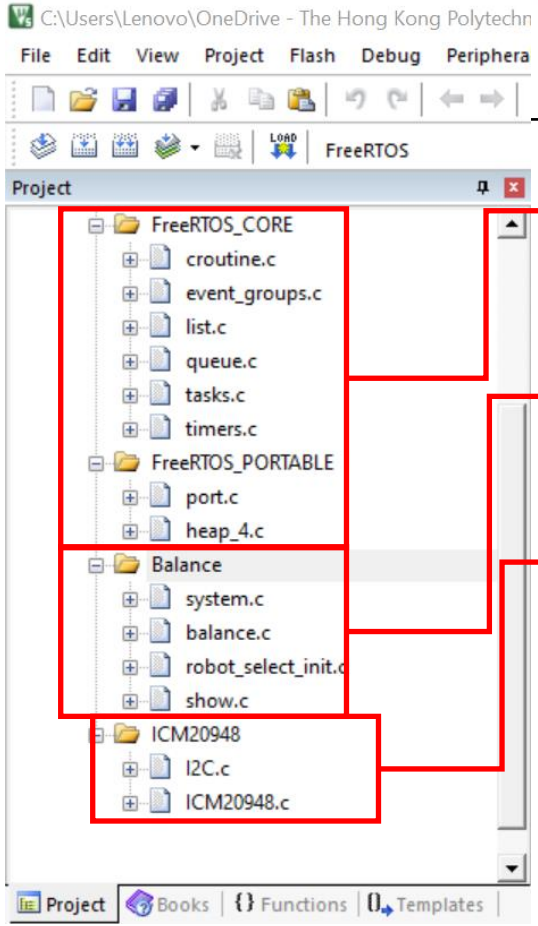
Chip configuration

Chip internal module configuration





'main.c' is the project entry point



General code function package,
defining data structure, timer, code scheduling, etc

Codes related to car movement,
motor control, sensor data collect and display

Communication Bus

Changing the car parameters

PID controller

STM32

source code

Kp

Ki

system.c* robot_select_init.c

```
16  Update£°2021-01-29
17
18  All rights reserved
19  ************************************************/
20
21  #include "system.h"
22
23  //Robot software fails to flag bits
24  //»úÆ÷ÈËÊÏ¼þÊ§ÄÜ±êÖ¾Î»
25  u8 Flag_Stop=1;
26
27  //The ADC value is variable in segments, depending on the number of car models. Currently there are 6 car models
28  //ADCÖµ·Ö¶Î±äÁ¿£¬¸ù¾Ý²»Í¬³µÐÍ¡¢²»Í¬Å
29  int Divisor_Mode;
30
31  // Robot type variable
32  //»úÆ÷ÈËÐÍºÅ±äÁ¿
33  //0=Mec_Car£¬1=Omni_Car£¬2=Akm_Car£¬3=Diff_Car£¬4=FourWheel_Car£¬5=Tank_Car
34  u8 Car_Mode=0;
35
36  //Servo control PWM value, Ackerman car special
37  //¶æ»ú¿ØÖÆPWMÖµ£¬°¢¿ËÃüÐ¡³µ×¨ÓÃ
38  int Servo;
39
40  //Default speed of remote control car, unit: mm/s
41  //Ò£¿ØÐ¡³µµÄÄ¬ÈÏËÙ¶È£¬µ¥Î»£°mm/s
42  float RC_Velocity=1000;
43
44  //Vehicle three-axis target moving speed, unit: m/s
45  //Ð¡³µÈýÖáÄ¿±êÔË¶¯ËÙ¶È£¬µ¥Î»£°m/s
46  float Move_X, Move_Y, Move_Z;
47
48  //PID parameters of Speed control
49  //ËÙ¶È¿ØÖÆPID ²ÎÊý
50  float Velocity_KP=300,Velocity_KI=300;
51
52  //Smooth control of intermediate variables, dedicated to omni-directional moving cars
53  //Æ½»¬¿ØÖÆÖÐ¼ä±äÁ¿£¬×¨Ïò¶à¶¯Ð¡³µ×¨ÓÃ
54  Smooth_Control smooth_control;
55
56  //The parameter structure of the motor
57  //µç»úµÄ²ÎÊý½á¹¹Ìå
58  Motor_parameter MOTOR_A,MOTOR_B,MOTOR_C,MOTOR_D;
59
60  /************ Ð¡³µPÎ»ÖÃ±Õ»·¿ ************************/
```

A demonstration vedio also on github

main ▾ Autonomous-Car-Control-Project / STM32 / Go to file Add file ▾ ···

AAE3004 Add files via upload 617aaf1 27 days ago History

..

1. Download the program via USB	Add files via upload	28 days ago
AAE3004 - MANUAL FOR ACCESING STM32 SOURCE CO...	Add files via upload	28 days ago
AAE3004 - PID controller demonstration.mp4	Add files via upload	28 days ago
CP2102Driver.zip	Add files via upload	28 days ago
FlyMcu.exe	Add files via upload	27 days ago
STM32 source code_F407VET6_2021.09.27.zip	Add files via upload	28 days ago
User Guide.pdf	Add files via upload	28 days ago