brain

Generated by Doxygen 1.8.11

Contents

Chapter 1

Module Index

1.1 Modules

Here is a list of all modules:

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

Module Documentation

4.1 ROS related variables

Variables

- ros::NodeHandle HubertBrain::nh_
- ros::Subscriber HubertBrain::face_found_sub
- ros::Subscriber HubertBrain::feedback_sub
- ros::Publisher HubertBrain::neck_pan_loop_
- ros::Publisher HubertBrain::neck_tilt_loop_
- ros::Publisher HubertBrain::body_loop_
- ros::Publisher HubertBrain::robot_elbow_
- ros::Publisher HubertBrain::robot_shoulder_
- ros::Publisher HubertBrain::fire_gun_
- ros::Publisher HubertBrain::say_hello_
- ros::Publisher HubertBrain::start_interrogation_
- ros::Publisher HubertBrain::access_details_
- ros::Publisher HubertBrain::feedback_pub_
- ros::ServiceClient HubertBrain::terminalClient

4.1.1 Detailed Description

In this group all the variables related to ROS will be define

4.1.2 Variable Documentation

4.1.2.1 ros::Publisher HubertBrain::access_details_ `[private]`

Publish the output from the interrogation (To Agent/ Type brain::Access)

4.1.2.2 ros::Publisher HubertBrain::body_loop_ `[private]`

Publish base angle for the base of the robot (To Servo/ Type UInt16)

4.1.2.3 ros::Subscriber HubertBrain::face_found_sub [private]

This is the subscriber for the face found value. (From face_detection node)

4.1.2.4 ros::Publisher HubertBrain::feedback_pub_ [private]

Publish the current status of the system (To Agent/ Type brain::Feedback)

4.1.2.5 ros::Subscriber HubertBrain::feedback_sub [private]

This is the subscriber for the feedback message. (From Agent node)

4.1.2.6 ros::Publisher HubertBrain::fire_gun_ [private]

Publish the fire gun command (To Servo/ Type Bool)

4.1.2.7 ros::Publisher HubertBrain::neck_pan_loop_ [private]

Publish pan angle for the neck (To Servo/ Type UInt16)

4.1.2.8 ros::Publisher HubertBrain::neck_tilt_loop_ [private]

Publish tilt angle for the neck (To Servo/ Type UInt16)

4.1.2.9 ros::NodeHandle HubertBrain::nh_ [private]

ROS node handler

4.1.2.10 ros::Publisher HubertBrain::robot_elbow_ [private]

Publish elbow angle for the arm of the robot (To Servo/ Type UInt16)

4.1.2.11 ros::Publisher HubertBrain::robot_shoulder_ [private]

Publish shoulder angle for the arm of the robot (To Servo/ Type UInt16)

4.1.2.12 ros::Publisher HubertBrain::say_hello_ [private]

Publish the initial welcome command (To Agent/ Type Bool)

4.1.2.13 ros::Publisher HubertBrain::start_interrogation_ [private]

Publish the interrogation dialogue trigger command (To Agent/ Type Bool)

4.1.2.14 ros::ServiceClient HubertBrain::terminalClient [private]

ROS Service to handle the terminal application

4.2 General variables

Variables

- int HubertBrain::pan_ub
- int HubertBrain::pan_lb
- int HubertBrain::pan_step_size
- int HubertBrain::tilt_ub
- int HubertBrain::tilt_lb
- int HubertBrain::tilt_step_size
- int HubertBrain::body_ub
- int HubertBrain::body_lb
- int HubertBrain::body_step_size
- int HubertBrain::ppl_passby_count
- int HubertBrain::current_body_angle
- bool HubertBrain::face_found
- bool HubertBrain::previous_face_found
- bool HubertBrain::state_changed
- bool HubertBrain::access_granted
- bool HubertBrain::face_init
- std::vector< uint > HubertBrain::panAngles_
- std::vector< uint > HubertBrain::tiltAngles_
- std::vector< uint > HubertBrain::bodyAngles_
- RobotState HubertBrain::robotState = RobotState::Idling
- WarningState HubertBrain::warningState = WarningState::NoWarning
- WarningState HubertBrain::previousWS = WarningState::NoWarning

4.2.1 Detailed Description

In this group all the general variable will be define

4.2.2 Variable Documentation

4.2.2.1 bool HubertBrain::access_granted [private]

Decision variable for FeedbackWaitingState and DecisionMaking state

4.2.2.2 int HubertBrain::body_lb [private]

Lower bound for the move base angle (user define from launch file)

4.2.2.3 int HubertBrain::body_step_size [private]

Step size for the move base angle (user define from launch file)

4.2.2.4 int HubertBrain::body_ub [private]

Upper bound for the move base angle (user define from launch file)

4.2.2.5 std::vector<uint> HubertBrain::bodyAngles_ [private]

Calculated body angle sequence according to the parameters provided by the launch file

4.2.2.6 int HubertBrain::current_body_angle [private]

Track the current body angle. (The move base angle)

4.2.2.7 bool HubertBrain::face_found [private]

Boolean to represent the face found status

4.2.2.8 bool HubertBrain::face_init [private]

Initial state of the face detection/tracking

4.2.2.9 int HubertBrain::pan_lb [private]

Lower bound for the pan angle (user define from launch file)

4.2.2.10 int HubertBrain::pan_step_size [private]

Step size for the pan angle (user define from launch file)

4.2.2.11 int HubertBrain::pan_ub [private]

Upper bound for the pan angle (user define from launch file)

4.2.2.12 std::vector<uint> HubertBrain::panAngles_ [private]

Calculated pan angle sequence according to the parameters provided by the launch file

4.2.2.13 int HubertBrain::ppl_passby_count [private]

The counter to check if the current tracking person disappear or not (if the counter is zero then its the same person, else few people have already pass the robot, when the robot in a certain transition state)

4.2.2.14 bool HubertBrain::previous_face_found [private]

Detect the changes in the face found (current face disappear or not)

4.2.2.15 WarningState HubertBrain::previousWS = WarningState::NoWarning [private]

To keep track of the Warning State

4.2.2.16 RobotState HubertBrain::robotState = RobotState::Idling [private]

Initial RobotState

4.2.2.17 bool HubertBrain::state_changed [private]

RobotState has change or not

4.2.2.18 int HubertBrain::tilt_lb [private]

Lower bound for the tilt angle (user define from launch file)

4.2.2.19 int HubertBrain::tilt_step_size [private]

Step size for the tilt angle (user define from launch file)

4.2.2.20 int HubertBrain::tilt_ub [private]

Upper bound for the tilt angle (user define from launch file)

4.2.2.21 std::vector<uint> HubertBrain::tiltAngles_ [private]

Calculated tilt angle sequence according to the parameters provided by the launch file

4.2.2.22 WarningState HubertBrain::warningState = WarningState::NoWarning [private]

Initial WarningState

Chapter 5

Class Documentation

5.1 HubertBrain Class Reference

```
#include <brain.h>
```

Public Types

- enum RobotState {
  Idling, Tracking, Interrogation, FeedbackWaitingState,
  DecisionMaking }

  ENUM Defining tracking state available.
- enum WarningState { NoWarning, InitialWarning, SecondWarning, FinalWarning }

  ENUM Defining warning states available.

Public Member Functions

- HubertBrain ()

  A Constructor for the class.
- void execute (void)

  The executor node.

Private Member Functions

- void faceFoundCallback (const std_msgs::Bool &msg)

  This callback will trigger with the face found details.
- void feedbackCallback (const brain::Feedback &msg)

  This callback will trigger with the face found details.
- void checkRobotStates ()

  This function will be used to switch the RobotState.
- void idlingLoop ()

  This function will be used to perform the idling state.
- void trackingStateLoop ()

  This function will be used to perform face tracking.
- void handleInterrogation ()

This function will be used control the terminal application.
- void feedbackWaiting ()

  This function handle the waiting times.
- void takeDecision ()
- void resetSystem ()

  This function handles the resetting process.
- std::vector< uint > getPanAngles (int lb, int ub, int stepSize)

  Calculated pan angle sequence.
- std::vector< uint > getTiltAngles (int lb, int ub, int stepSize)

  Calculated tilt angle sequence.
- std::vector< uint > getBodyAngles (int lb, int ub, int stepSize)

  Calculated body angle sequence.

Private Attributes

- ros::NodeHandle nh_
- ros::Subscriber face_found_sub
- ros::Subscriber feedback_sub
- ros::Publisher neck_pan_loop_
- ros::Publisher neck_tilt_loop_
- ros::Publisher body_loop_
- ros::Publisher robot_elbow_
- ros::Publisher robot_shoulder_
- ros::Publisher fire_gun_
- ros::Publisher say_hello_
- ros::Publisher start_interrogation_
- ros::Publisher access_details_
- ros::Publisher feedback_pub_
- ros::ServiceClient terminalClient
- int pan_ub
- int pan_lb
- int pan_step_size
- int tilt_ub
- int tilt_lb
- int tilt_step_size
- int body_ub
- int body_lb
- int body_step_size
- int ppl_passby_count
- int current_body_angle
- bool face_found
- bool previous_face_found
- bool state_changed
- bool access_granted
- bool face_init
- std::vector< uint > panAngles_
- std::vector< uint > tiltAngles_
- std::vector< uint > bodyAngles_
- RobotState robotState = RobotState::Idling
- WarningState warningState = WarningState::NoWarning
- WarningState previousWS = WarningState::NoWarning

5.1.1 Member Enumeration Documentation

5.1.1.1 enum HubertBrain::RobotState

ENUM Defining tracking state available.

Enumerator

Idling Initial state - Contains the Idling loop
Tracking Tracking state - Will move to this when a face is detected
Interrogation Interrogation state - Will handle opening the terminal application
FeedbackWaitingState Feedback waiting - To handle the waiting after interrogation
DecisionMaking DecisionMaking - Make the decision based on the interrogation output

5.1.1.2 enum HubertBrain::WarningState

ENUM Defining warning states available.

Enumerator

NoWarning No warning - In the first 3 robots state the warningState is set to this
InitialWarning Initial warning - Define the initial warning after interrogation
SecondWarning Second warning
FinalWarning Final warning

5.1.2 Constructor & Destructor Documentation

5.1.2.1 HubertBrain::HubertBrain ( )

A Constructor for the class.

Will be using to initialize initial values of the user variables and the ROS parameters, i.e Subscribers, Publishers and the lanuch file params

5.1.3 Member Function Documentation

5.1.3.1 void HubertBrain::checkRobotStates ( ) `[private]`

This function will be used to switch the RobotState.

In This function, it will be constantly check the current robot state. This will run in the main thread of the ROS node

5.1.3.2 void HubertBrain::execute (void)

The executor node.

This node will be call from the ROS NODE main and will start the main ROS thread for publishers and subscribers. Node will run with 10 Hz rate.

5.1.3.3 void HubertBrain::faceFoundCallback (const std_msgs::Bool & *msg*) [private]

This callback will trigger with the face found details.

In this callback, the data coming from the face_detection node will be analyzed. According to this boolean data message the RobotState may change.

5.1.3.4 void HubertBrain::feedbackCallback (const brain::Feedback & *msg*) [private]

This callback will trigger with the face found details.

In this callback, the data coming from the face_detection node will be analyzed. According to this boolean data message the RobotState may change.

5.1.3.5 void HubertBrain::feedbackWaiting ( ) [private]

This function handle the waiting times.

This function handles the decision making state.

In This function, the waiting time between the Agent and the current robot status will be handled. The state FeedbackWaiting was introduced to handle the the transition section from Interrogation to DecisionMaking, as there will be few sub processes running in the back-ground.

In This function, DecisionMaking state will be handle. The state DecisionMaking will be responsible to handle the last phase of the operation. Upon the decision the robot will publish date to the servo node or it will simply do a reset

5.1.3.6 std::vector< uint > HubertBrain::getBodyAngles (int *lb,* int *ub,* int *stepSize*) [private]

Calculated body angle sequence.

The values will be according to the parameters provided by the launch file

Parameters

in	*lb*	Lower bound
in	*ub*	Upper bound
in	*stepSize*	step size of the angle

Returns

A vector with body angles that should be published in the idling loop

5.1.3.7 std::vector< uint > HubertBrain::getPanAngles (int *lb,* int *ub,* int *stepSize*) [private]

Calculated pan angle sequence.

The values will be according to the parameters provided by the launch file

Parameters

in	*lb*	Lower bound
in	*ub*	Upper bound
in	*stepSize*	step size of the pan angle

Returns

A vector with pan angles that should be published in the idling loop

5.1.3.8 std::vector< uint > HubertBrain::getTiltAngles (int *lb*, int *ub*, int *stepSize*) [private]

Calculated tilt angle sequence.

The values will be according to the parameters provided by the launch file

Parameters

in	*lb*	Lower bound
in	*ub*	Upper bound
in	*stepSize*	step size of the tilt angle

Returns

A vector with pan angles that should be published in the idling loop

5.1.3.9 void HubertBrain::handleInterrogation ( ) [private]

This function will be used control the terminal application.

In This function, interrogation will be carried out. This will handle the servo movement and the terminal application. Upon the terminal application output the function will the details to Agent. Will also initialize the tracking

5.1.3.10 void HubertBrain::idlingLoop ( ) [private]

This function will be used to perform the idling state.

In This function, it will publish data to servo package to move the neck and body in a sequence. This will have the access to the global variables panAngles_, tiltAngles_ and bodyAngles_. Will run in the a different thread so that it will not interrupt the main ROS node

5.1.3.11 void HubertBrain::resetSystem ( ) [private]

This function handles the resetting process.

In This function, the system will be resetted. All the decision variables will be set to the initial values and the robot will also move to its initial positions

5.1.3.12 void HubertBrain::takeDecision ( ) [private]

5.1.3.13 void HubertBrain::trackingStateLoop ( ) [private]

This function will be used to perform face tracking.

In This function, face tracking will be performed. This function will send commands to the trackFace node to perform its task. Once the waiting is over the function will decide whether or not to change to the interrogation state. Will run in the a different thread so that it will not interrupt the main ROS node

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

- brain.h
- brain.cpp

5.2 ROSFaceDetection Class Reference

The ROS wrapper class for Hubert's Face Detection application.

```
#include <brain.h>
```

5.2.1 Detailed Description

The ROS wrapper class for Hubert's Face Detection application.

Perform face detection and face tracking with the help of an external library FaceTracking under ROS environment

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

- brain.h

Chapter 6

File Documentation

6.1 brain.h File Reference

This is the header file of the brain.cpp, which was implemented to handle all the operation in the Hubert system. This works as the centralized hub to control all t the other nodes. The face detection/tracking, terminal application and the Agent application.

```
#include "ros/ros.h"
#include "ros/package.h"
#include <std_msgs/Bool.h>
#include <std_msgs/Int32.h>
#include <std_msgs/UInt16.h>
#include <std_msgs/UInt16MultiArray.h>
#include "brain/RequestTerminal.h"
#include "brain/Access.h"
#include "brain/Feedback.h"
```

Classes

- class HubertBrain

6.1.1 Detailed Description

This is the header file of the brain.cpp, which was implemented to handle all the operation in the Hubert system. This works as the centralized hub to control all t the other nodes. The face detection/tracking, terminal application and the Agent application.

Author

Fredrik Lagerstedt
Zhanyu Tuo
Terje Stenstrom
Nipun C. Gammanage

Date

Initial release - October/2019

6.2 brain.cpp File Reference

```
#include "brain.h"
```

6.3 brain_node.cpp File Reference

```
#include "brain.h"
```

Functions

- int main (int argc, char **argv)

6.3.1 Function Documentation

6.3.1.1 int main (int *argc,* char ** *argv*)

Index