without modifying main application

Code.

(Catkin-Concate-Pkg plugimlib-Calculator (Plugialib soscpp stdamsgo Step 1: Concading Calculation-base header main purpose of this file is to declare functions knotheds that are commonly word by the plugions. namespare calculdon-boso L> Class -> calc-functions 4> Member Petians > get-mumbers Lo operation Stop2: Cooling Calculaton-plugins header => Main furitions of this * calculator Plugims, which are named as Add , Sub, Mul, and DIV. Stile is to define (Complete function of the The state of the s Marine Andrews in the State of the State

Park Park Stop3 - Exposting plugins wing calculation - Plugins-CPP => In ooder to load the class of Plugins dynamically, we have to expost each Class using a special macro Called PLUGINLIB_EXPORT_CLASS. # Include Lplugimlib/class_list_mocrosity / PLUGINLIB-EXPORT_CLASS (Calculator -plugins: Add, calculatos-bese :: calc_function); Stepy-Implementing plugin loader using calculator - loader CPP => The plugh loader node loads each Plugin and Imputs the number to each plugin and fetch's the onesult From the plugin. S # include 2boost/Shand_ptor.hpp> # include 2 Plugimlib/class-loader: b) Pluginlib: Class Loaden & calculation -base: (delications)

Calc_loader ("Pluginlib_Calculdos"

, "Calculdon-base: calc-functions");

boost: Shaned-pton & Calculation-buse :: calc-functions > add = Calc-loader. Consale Instance ("Pluginlib_ Calculdon/Add"); Step 5: Concading plugin description file : Calculator - Plugia.xml (liborary palh="lib/libpluginlib-Calculator") LClas mani = "pluginlib_calculatos /Add" tope = " calcaldon - plusins :: Add" bers-class-tepe=" Calculatos-base: calc-fullor") <clescription> This is cold plugin. Lidescription> 2/library> Stop 6: Registering plugin with the ROS Packago System >> For pluginlib to fine all plughes based Parkages in the 1205 System, we should e-point the plugin description file inside Perkege-mil.

=> If we do not include this plugin Sto , the ROS system wond And the plugins incide the perkage. 990 <or post> Ste < Plugin 11b_Calculator plugin = "\$[Pricfix] /calculator-plugins. m/"/> 9 </r>

build_dopond> pluginlib_calculation =7 2/build-depond> < onun-depend> plugimlib-calculation 1/som-depond Stop7- Editing the CMakeList. test file add_library (plugimlib-calculator Sonc/calculatos-plugims.cpp) => target-link-library (pluginlib-calculation \${Codkin_LIBRARIES}) = add-executable (Calculator-loader). tanget-link-libraries (Calculdosnilocden \$ (catkin -LIBRARIE))

Stop 8: Querying the list of pluging in a

grospale plugins -- attails = plugin pluginlib

Step 9: Running the plugin loader grosonum pluginlib-calculation adalah loader

* Understanding ROS modelets

=7 Nodelets are a type of ROS mode that are designed to onun multiple modes in a Single process, with each node onuning as a thrich.

Communicate with externed modes too.

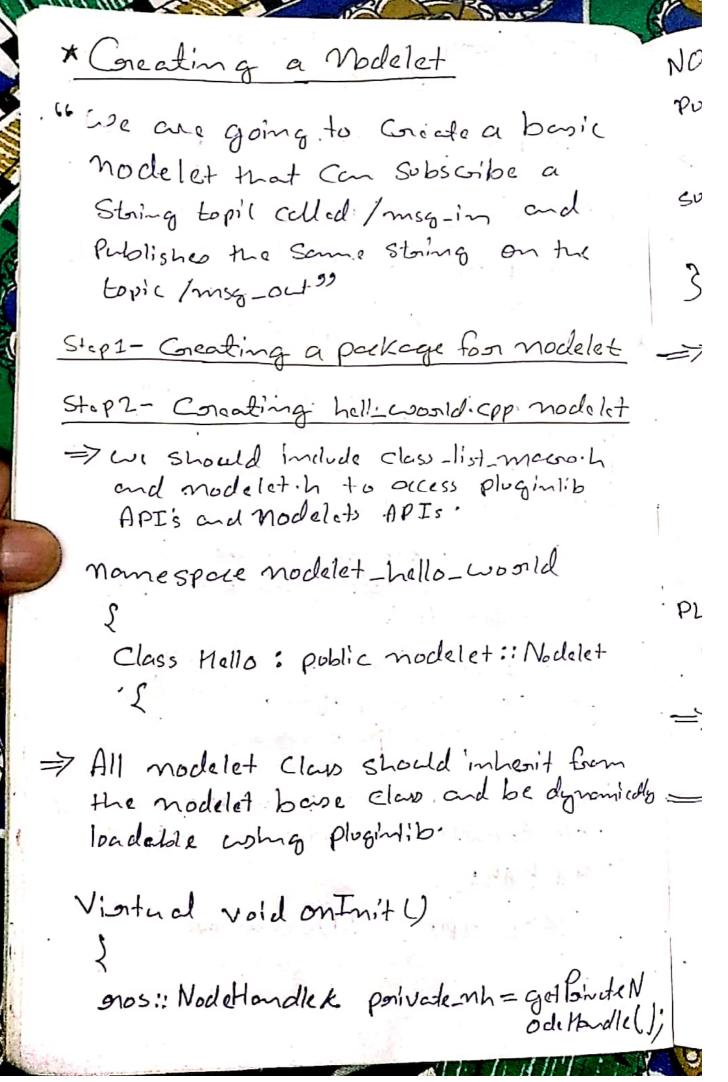
=> In nodelets, we dynamically load each class as a plugh, which has a Separate name space.

=> Nodelets are used when the Volume of data transferred between nodes are very high.

example => 30 senson

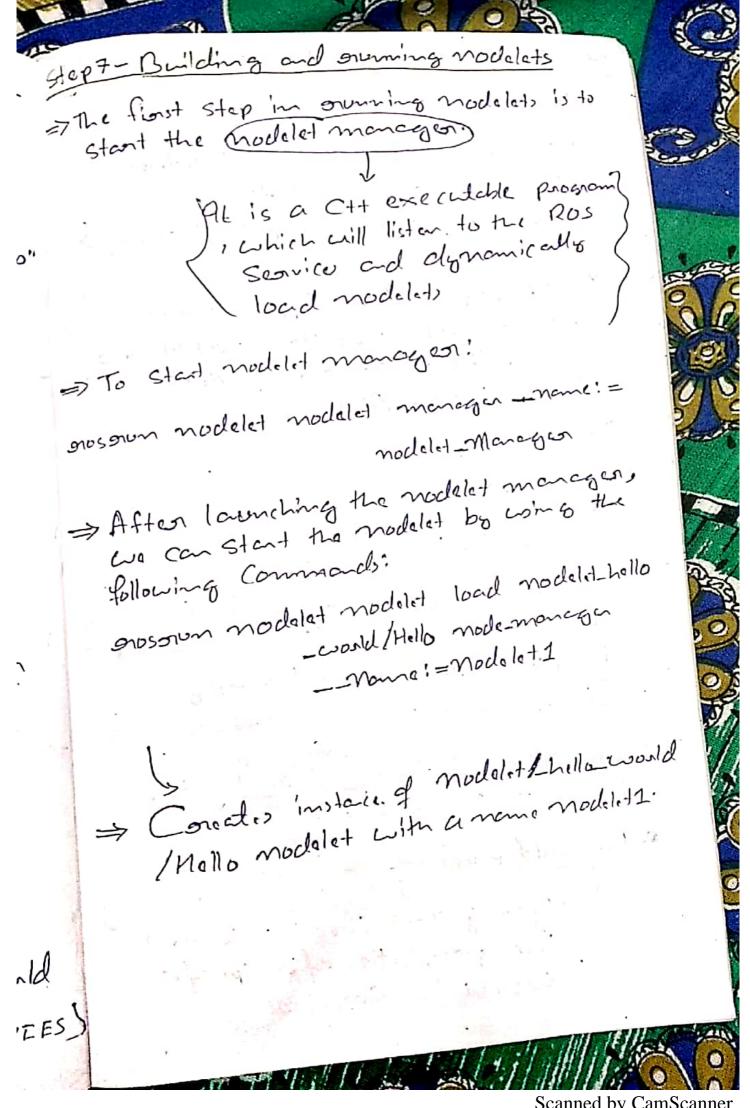
OID

Scanned by CamScanner



NODELET-DEBUG ("Initialized the Nodelot") pub = porivete_nh. adventise <std_msso: Stohes (hms 6-00",5); Sub= porivate_mh. subscribe ("msq_in", 5, LHEllo: callbook, this); 131 -... =7 This is the imitilization function of a modelet. = Anside this function we are creating a node handle object, topic publisher and Subsciber on the topic mig-out and msg-in suspectively. PLUGINLIB_EXPORT_CLASS (nodelet_hello_world: Hello, modelet: Nodelet); => Here we are exposting the Hello esa plugim for the dynamic loading.

Step 4 - Coneating plugin description file step hello- woold xml < libraro path = "libradelet-hello-woorld"> < class name = "nodelet-hello-world! Mollo" Eype = "nodelot-hello-woorld: HEllo" base-class-type="modelet:: Nodelet"> < doscolption> A node to orapublish a message 9705 Z/desconiption> 4/class> </11/20 Comod/1/> Steps - Adding the expost tog in Pakago.xml We need to odd the exposit tag in Parkege: xml and also add build and our dependincies. Stepb-Editing CMakeList txt add-liberaries (nodelet-hello-wasld Sonc/hillo-worldicpp) tanget - link-libraries (nodelet-hallo-world \$[Cetkin-LIBRARIES)



Step 8 - Coreating laurch files foor modelets

< launch>

<node pkg="nodelet" type="nodelet"

nome = "stadalone-nodelet"

angs = "managen" output = "scoreen"/>

Znode pkg="nodelet" Expe="nodelet"

nome=" Eest" angs="load nodelet_hello
- world/hello stendatoni_nodelet"

oudput="screen"/>

4launch>

* Gazebo Plugins

- => Gazabo plugins help us to control
 the properties, and even the way
 Gazabo gruns.
- => We can Mainly classify the Plugins as follows:
 - Don't plugin

 Susing this we can Control

 the properties of a specific

 would in Gazobo

