System Synthesis Supplementary Data

Jason Ziglar Ryan Williams Alfred Wicks

I. Introduction

In order to explore the capability of automated system synthesis to compose a variety of novel systems from a common set of components, an experiment simulating the design of a mobile search and rescue robotic system is developed. As an automated system synthesis experiment, this experiment explores how an automated system can generate the structure of an design which can be implemented with a defined set of components available to satisfy the design; the components used in these designs are taken as an input into the system. Thus, this document describes the defined components used in this experiment, as well as how the experiments were organized and executed, with the system performance both in terms of computational requirements and resulting designs reported. The formulation of the system synthesis approach used in these experiments are described in work referencing this document.

This experiment defines a variety of common hardware and software components which may be useful for the implementation of the proposed robotic system in a variety of operating conditions. These components are based on describing components developed or used for real world components developed for a variety of previous robotics research projects. This experiment also defines a set of functional capabilities which may be included in the system, as well as a set of environmental parameters which may impact how such functionality must be implemented. These components include some components which provide similar functional capabilities with varying non-functional requirements (e.g. differing types of connection standards, differing computational resource budgets/requirements) in order to flesh out the design space such that trade-offs between selections exist.

II. Setup and Results

A. Hypothetical Mission

These experiments consider the design and development of a hypothetical mobile robot for search and rescue experiments. This system is meant to represent a fairly generic search and rescue mission - the specific details of search is abstracted away to "detect and localize some object of interest", while rescue is simplified to "have the ability to manipulate the localized object of interest." Such a robot has several subsystems which are required for operation: a mobile base capable of navigating an environment, the ability to sense objects of interest for rescue, and the ability to perform the aforementioned rescue. Additionally, software components can be included which provide the ability to perform some amount of this mission autonomously: path planners for the mobile base or manipulator, localization for integration of sensor data into a common frame of reference, or higher level planning for ensuring an area is fully searched. As an iterative design process, this experiment replicates the design of this type of system in a typical research environment - components are modularized such that some subsystems can be composed into systems which provide a subset of the full system's capability. This allows reuse between unrelated systems, as well as testing and iterative integration of components.

B. Functional Requirements and Operational Environments

The functional capabilities described in this experiment aim to describe the high level functional requirements described in the hypothetical mission scenario. Functional decomposition can be done in a variety of ways depending on the components under consideration, problem definition, and personal preference, leaving no particular decomposition as correct or incorrect. For the purposes of this experiment, the functional decomposition selected is chosen based on a few heuristics driven by the goals of the experiment, as opposed to the underlying design problem. First, the functional requirements should correspond to ones that are intuitive and easily understood in the context of designing robotic systems, in order to more easily communicate the relationship between functional requirements and system design. Second, the requirements should decompose the system in a fashion which allows for describing functionally coherent subsets of the complete system. This allows demonstrating the ability of the system synthesis framework to generate structure in a range of complexities, as well as measuring how individual components and capabilities impact the performance of the system. Third, these requirements should represent both binary valued (e.g. functionality which is either present or missing) as well as real valued (e.g. the range at which tracking can happen) in order to demonstrate the array of functional requirements which can be expressed. Finally, given the combinatoric nature of this experiment, the number of functional capabilities should be selected such that the size of the parameter space demonstrated is large enough to provide a realistic scenario, but still within the realm of reasonable analysis and description.

To meet these ends, the functionality defined for these missions is broken down into five separate components: target tracking, mobility, manipulation, and planning for either the manipulation or mobility components. While target tracking is a real valued functional requirement, representing the range at which tracking can happen, the other four values are binary. Several of the components defined also describe other functionality which can be provided, such as localization, which are required for other components to function, or could be explicit system requirements in their own right, but are not explicitly required in these experiments. Components can thus be defined as providing some number of these functional capabilities, while a system design can be described as requiring some number of these functional requirements.

For operational environments, these parameters aim to describe relevant parameters of the situation in which this system is designed to operate. While a real world environment involves many details which are irrelevant to a particular design problem, this experiment considers a subset of parameters which are all relevant to the considered design problems. For the given mission, the operating environment is parameterized by set of three parameters. The first parameter represents whether the environment is lit (e.g. daytime or nighttime), which impacts the sensors which might track an object of interest. The second parameter represents whether the mission is taking place outdoors, where GPS signals would be available for localization, or indoors, where alternate localization techniques are necessary (e.g. SLAM). Finally, the visual parameter represents whether the object of interest can be tracked via visual cues, or requires using a beacon based tracking approach. These parameters do not describe the full extent of a real world operating context, but provide sufficient variability to allow components to specialize in handling particular circumstances, demonstrating the ability for such a system to adapt structure to these limitations.

C. Component Library

The component library specified for these design problems describes the hardware and software elements which may be used to satisfy the specified design problem. The components included in this library are based on components available for use in common robotic software systems such as the Robot Operating System (ROS), which provides many of these components for development of systems like the given design scenario. In order to complete this experiment, components must exist which can produce a system as specified in the various configurations with at least one valid design, requiring a library with multiple components to address each functional requirement depending on environmental parameters. For some subsystems, such as tracking and localization, multiple components are defined which can provide a particular unit of functional capability, either to address the limitations of individual components to handle the full range of operational environments or to consider some number of potential options which may represent trade-offs in the design space. For instance, multiple inertial measurement units (IMU's) are defined in the library, representing different trade-offs in accuracy, cost, and computers included represent differing amounts of computational power, connectivity, underlying architecture (e.g. ARM and x86 computers) and cost. This aims to provide a design space which is non-trivial for a system synthesis problem to consider.

D. Experimental Setup

In order to measure the impact of functional requirements and operational environments, the experiment measures the performance of system synthesis in completing designs over a variety of these scenarios. With the set of components under consideration fixed for all trials, the functional requirements of the system and operational parameters are varied for each trial. Functional requirements for the system are varied across 10 possible designs, meant to represent variations of the system which might be encountered in an iterative development environment. At one end of the spectrum are the individual functional subsystems: teleoperating the mobile base or arm, or a fixed sensor package for detecting the object of interest. In the middle, systems with partial autonomy are defined as arm or mobile base with integrated planning, or these systems with the tracker integrated. Finally, the complete system is also considered, which includes driving, tracking, manipulation, and full autonomy. There are several variations in the partial system range which could be evaluated, but correspond to systems in which the functionality is not a coherent set (e.g. an arm with the planner for driving) which are left out of consideration. For operational environments, all permutations of the 3 parameters defined are considered in conjunction with the functional requirements. The environmental parameters have some impact on functional ones: for parameters which are not binary valued such as the tracker detection range, the values are selected based on the limits required by implementations available (e.g. if the only tracker which can work in an environment has a 3m functional range, the functional requirement is set to 3m.) This defines a set of 80 possible combinations of functional requirements and operational environments for this experiment.

With the configurations defined, the automated system synthesis framework is used to generate the specified problem and solve it over three trials, with the computational time required to generate the solution recorded. In addition, the solution generated in each trial is recorded and visualized in the included figures, for better understanding the solutions which are generated in each trial. These figures include both the hardware and software designs synthesized by the system, as well as the assignment of software components to the underlying hardware system. Assignments are visualized via the color coding between the two graphs for ease of inspection. Additionally, the reported timings are used to perform a second-degree ANOVA analysis to investigate the impact of each parameter on the overall synthesis performance.

III. Results

The quantitative results of these experiments described here encompass both the computational effort required to produce these solutions, as well as the designs generated by this framework.

A. Full ANOVA Results

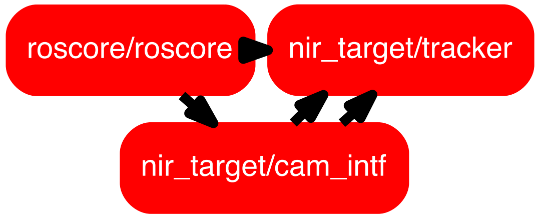
In order to estimate the relevance of the parameters representing the functional requirements and operational environment, a second-degree ANOVA analysis was performed. The analysis treated all parameters as binary valued where possible, with the tracking range being treated as a real-valued variable. The full results of this analysis is reported in Table I, computed using JMP Pro 12.

TABLE I
SECOND-DEGREE FACTORIAL ANOVA RESULTS.

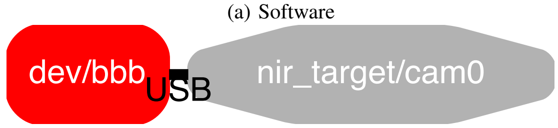
Term	**Estimate**	**Std Error**	**t-Ratio**	**Prob**$> \|t\|$
Intercept	4.661	0.06201	75.17	< 0.0001
Track	-0.02354	0.005964	-3.95	0.0001
Drive	0.3549	0.06294	5.64	< 0.0001
Plan-M	-0.03118	0.03087	-1.01	0.3136
Plan-A	-0.06229	0.03631	-1.72	0.0877
Arm	0.06603	0.06596	1.00	0.3179
Daylight	-0.1111	0.009582	-11.60	< 0.0001
Outdoors	0.1085	0.009582	11.33	< 0.0001
Visual	-0.5222	0.01026	-50.88	< 0.0001
Track*Drive	0.02628	0.03498	0.75	0.4532
Track*Plan-M	-0.006934	0.01467	-0.47	0.6369
Track*Plan-A	-0.04818	0.01467	-3.28	0.0012
Track*Arm	0.07632	0.03340	2.29	0.0233
Track*Daylight	0.01860	0.004807	3.87	0.0001
Track*Outdoors	-0.02196	0.004807	-4.57	< 0.0001
Track*Visual	-0.02191	0.005483	-4.00	< 0.0001
Drive*Plan-M	0	0	.	.
Drive*Plan-A	0.1259	0.06035	2.09	0.0381
Drive*Arm	0	0	.	.
Drive*Daylight	-0.02248	0.03289	-0.68	0.4950
Drive*Outdoors	0.1126	0.03289	3.42	0.0007
Drive*Visual	-0.1260	0.03118	-4.04	< 0.0001
Plan-M*Plan-A	0	0	.	.
Plan-M*Arm	0	0	.	.
Plan-M*Daylight	0.02113	0.02919	0.72	0.4700
Plan-M*Outdoors	0.003328	0.02919	0.11	0.9093
Plan-M*Visual	0.02445	0.03024	0.81	0.4198
Plan-A*Arm	0	0	.	.
Plan-A*Daylight	-0.01125	0.02919	-0.39	0.7004
Plan-A*Outdoors	0.0004866	0.02919	0.02	0.9867
Plan-A*Visual	-9.34e-5	0.03024	-0.00	0.9975
Arm*Daylight	-0.001081	0.03289	-0.03	0.9738
Arm*Outdoors	-0.03505	0.03289	-1.07	0.2877
Arm*Visual	0	0	.	.
Daylight*Outdoors	0.01596	0.009582	1.67	0.0972
Daylight*Visual	0.1348	0.01001	13.47	< 0.0001
Outdoors*Visual	-0.1318	0.01001	-13.17	< 0.0001

B. Variant Designs

The following figures provide the synthesized solutions generated in the course of the ANOVA analysis, representing the full combination of functional requirements and mission contexts available in the experiment. These figures are provided for visual inspection of the expressiveness available with system synthesis, as well as the common subsystem patterns present in these designs.



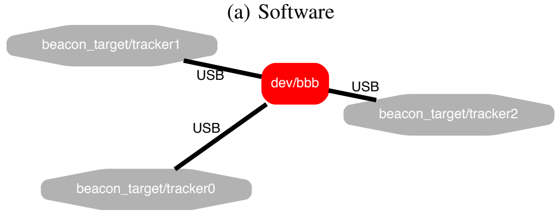
(a) Software



(b) Hardware

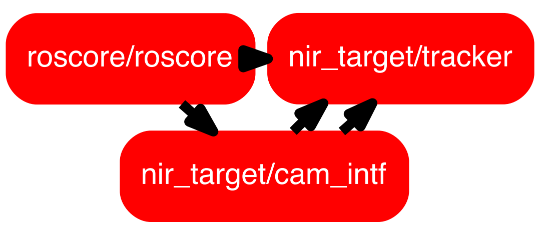
Fig. 1. Tracker Only Design: Night, Indoor, Visual Context

(a) Software

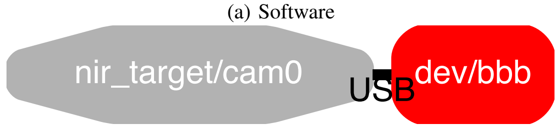


(b) Hardware

Fig. 2. Tracker Only Design: Night, Indoor, Beacon Context

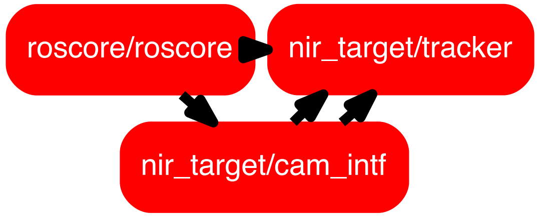


(a) Software

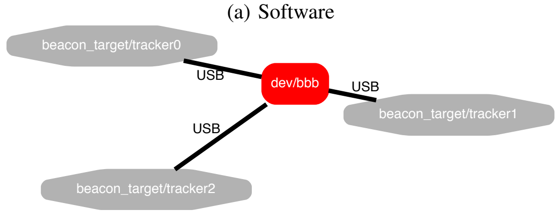


(b) Hardware

Fig. 3. Tracker Only Design: Night, Outdoor, Visual Context

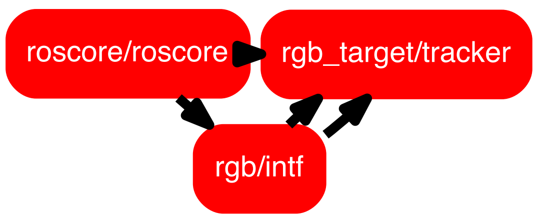


(a) Software

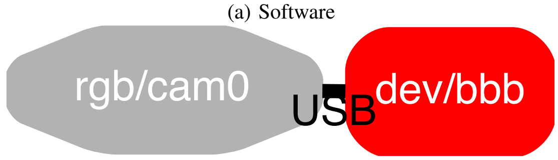


(b) Hardware

Fig. 4. Tracker Only Design: Night, Outdoor, Beacon Context

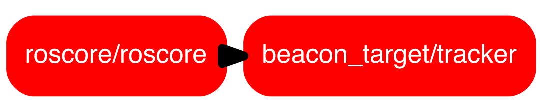


(a) Software

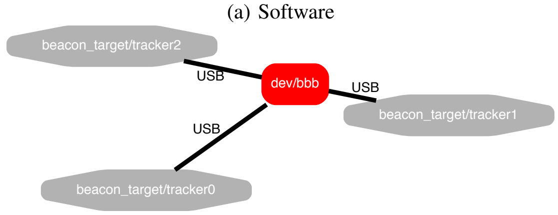


(b) Hardware

Fig. 5. Tracker Only Design: Day, Indoor, Visual Context

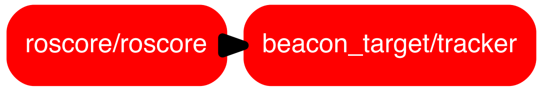


(a) Software

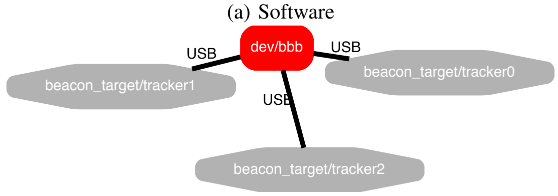


(b) Hardware

Fig. 6. Tracker Only Design: Day, Indoor, Beacon Context

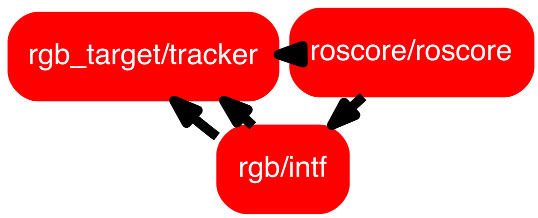


(a) Software

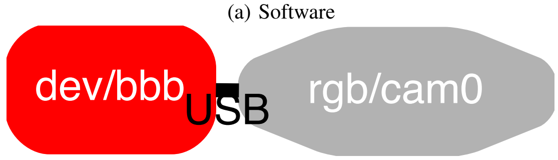


(b) Hardware

Fig. 7. Tracker Only Design: Day, Outdoor, Beacon Context

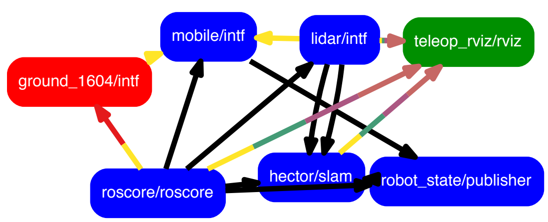


(a) Software

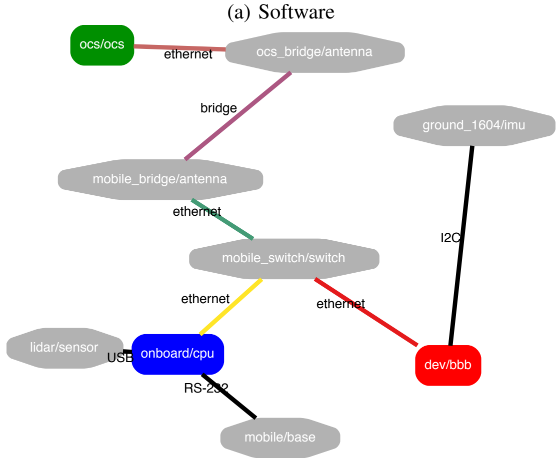


(b) Hardware

Fig. 8. Tracker Only Design: Day, Outdoor, Visual Context

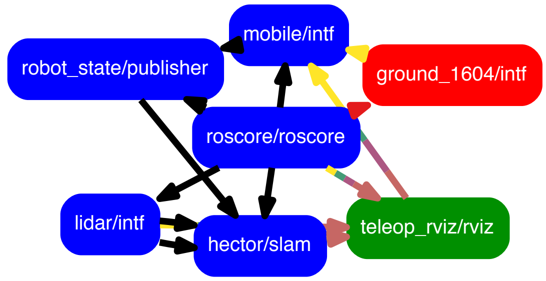


(a) Software

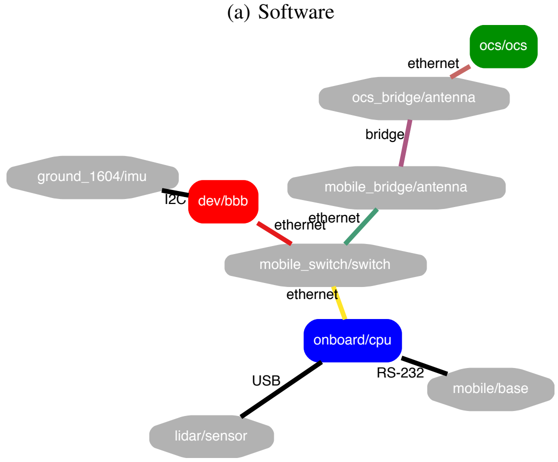


(b) Hardware

Fig. 9. Teleoperation Design: Night, Indoor, Visual Context

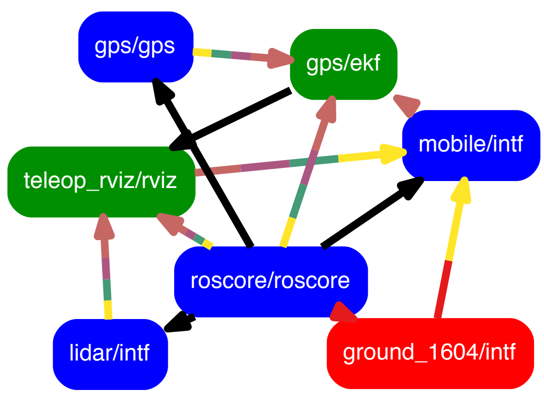


(a) Software

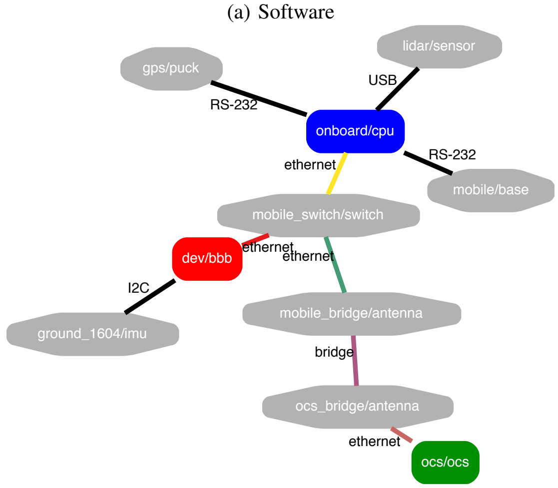


(b) Hardware

Fig. 10. Teleoperation Design: Night, Indoor, Beacon Context

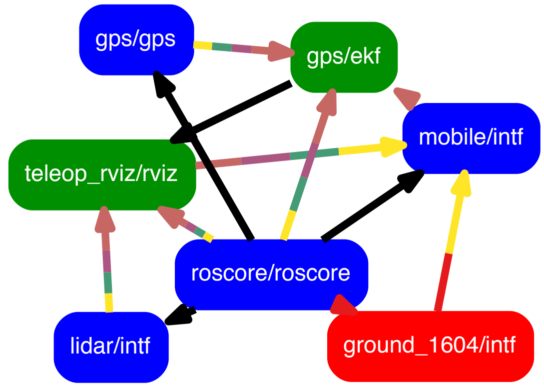


(a) Software

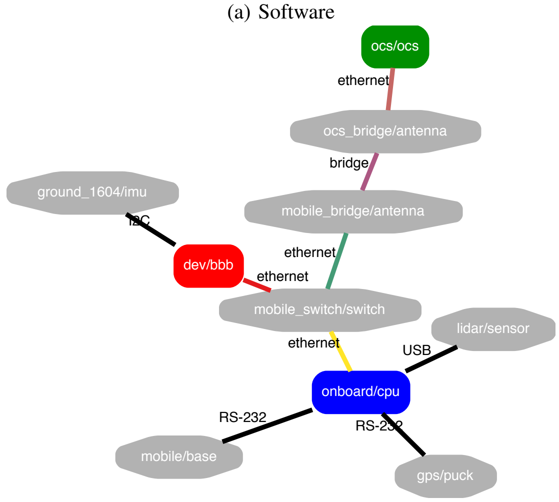


(b) Hardware

Fig. 11. Teleoperation Design: Night, Outdoor, Visual Context

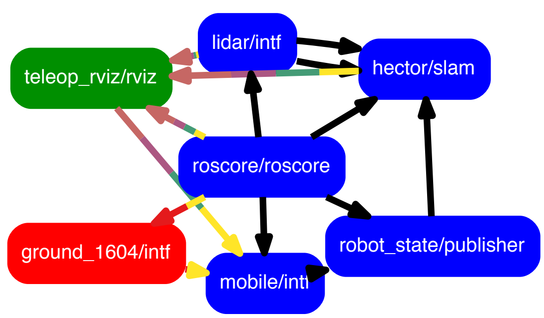


(a) Software

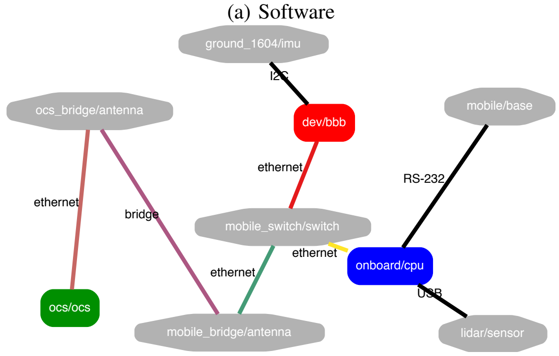


(b) Hardware

Fig. 12. Teleoperation Design: Night, Outdoor, Beacon Context

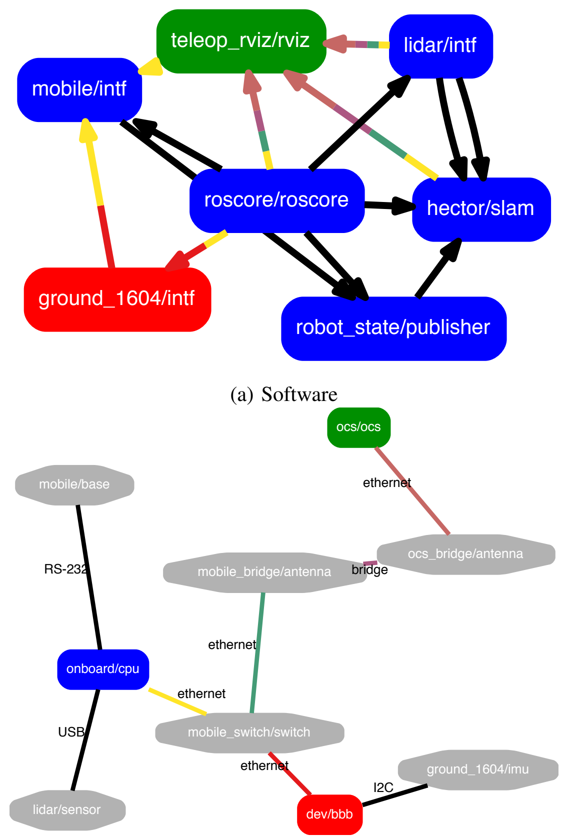


(a) Software



(b) Hardware

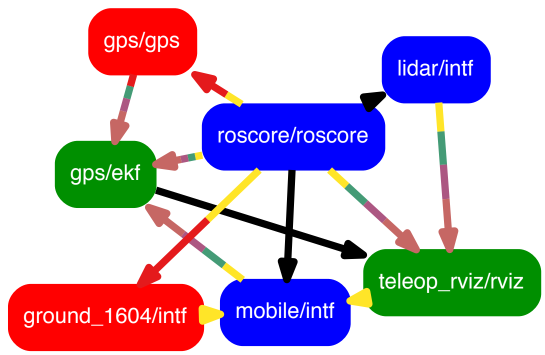
Fig. 13. Teleoperation Design: Day, Indoor, Visual Context



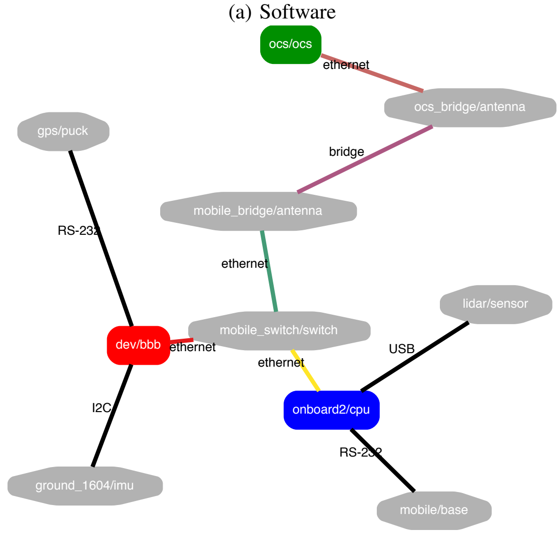
(a) Software

(b) Hardware

Fig. 14. Teleoperation Design: Day, Indoor, Beacon Context

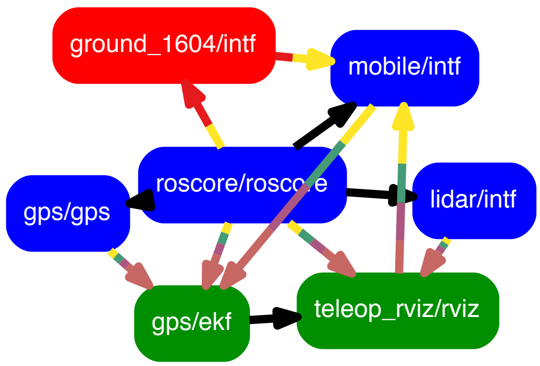


(a) Software

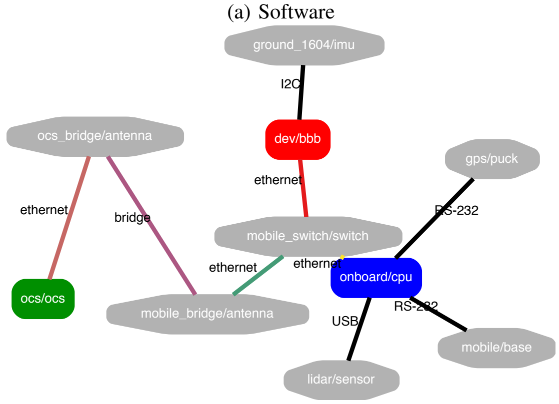


(b) Hardware

Fig. 15. Teleoperation Design: Day, Outdoor, Beacon Context

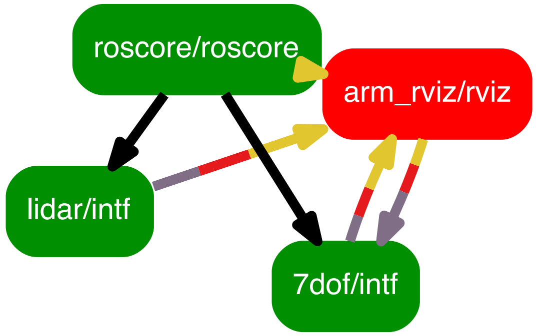


(a) Software

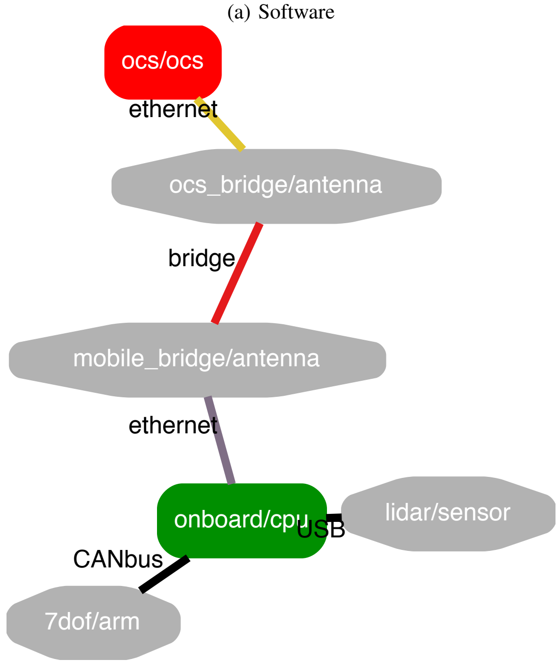


(b) Hardware

Fig. 16. Teleoperation Design: Day, Outdoor, Visual Context

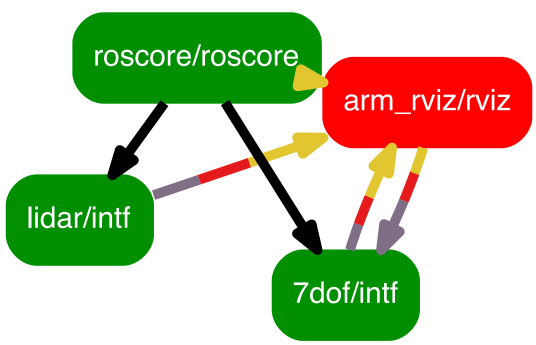


(a) Software

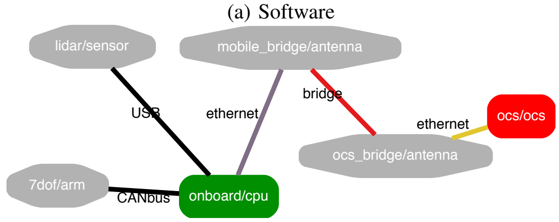


(b) Hardware

Fig. 17. Teleoperated Arm Design: Night, Indoor, Visual Context

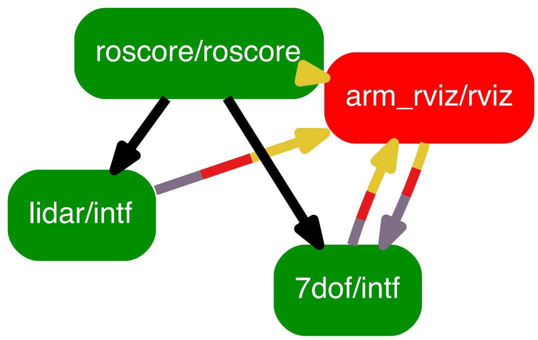


(a) Software

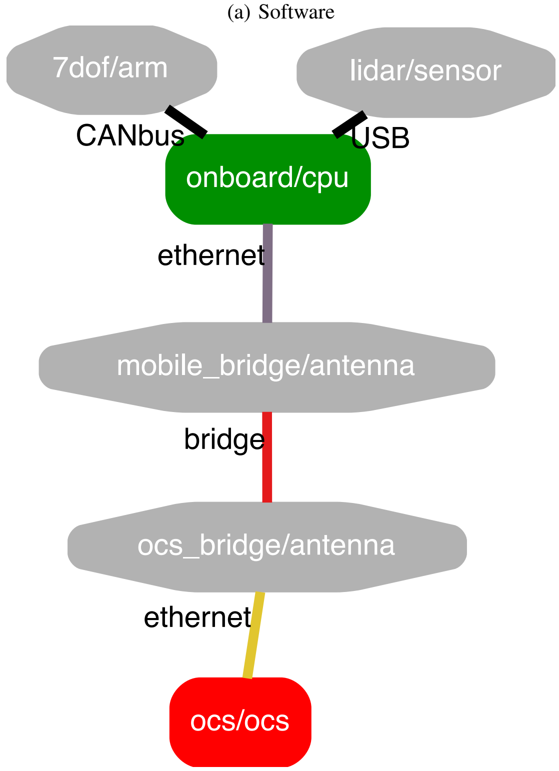


(b) Hardware

Fig. 18. Teleoperated Arm Design: Night, Indoor, Beacon Context

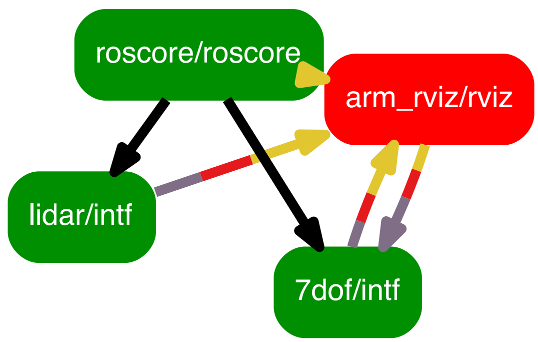


(a) Software

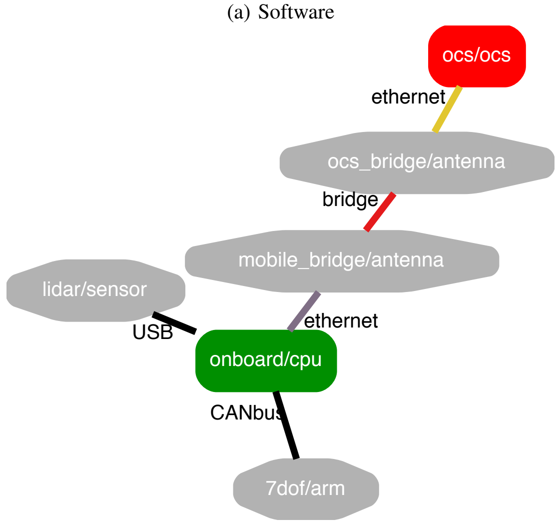


(b) Hardware

Fig. 19. Teleoperated Arm Design: Night, Outaoor, Visual Context

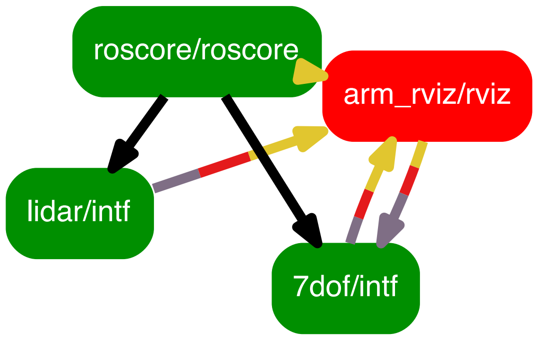


(a) Software

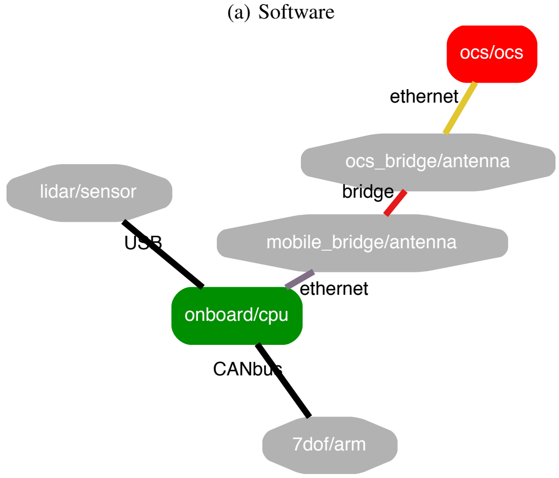


(b) Hardware

Fig. 20. Teleoperated Arm Design: Night, Outaoor, Beacon Context

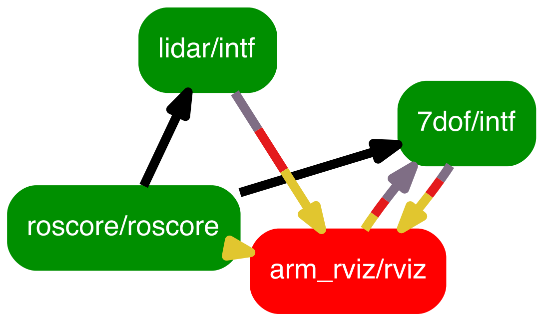


(a) Software

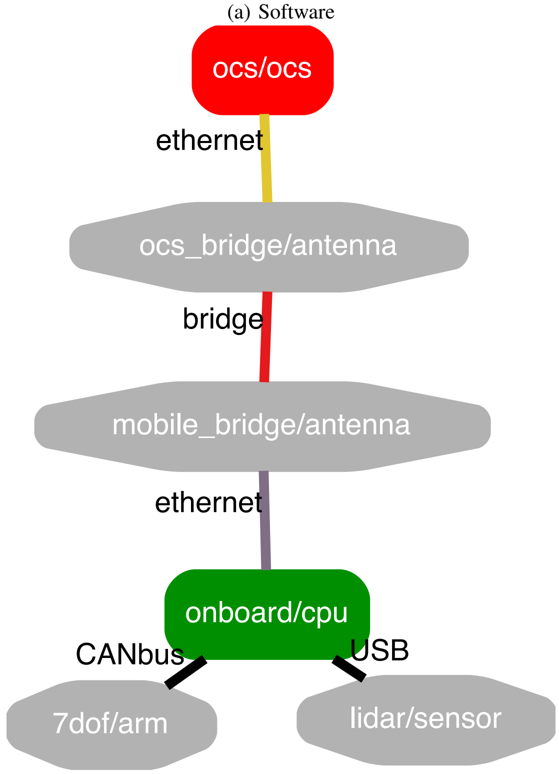


(b) Hardware

Fig. 21. Teleoperated Arm Design: Day, Indoor, Visual Context

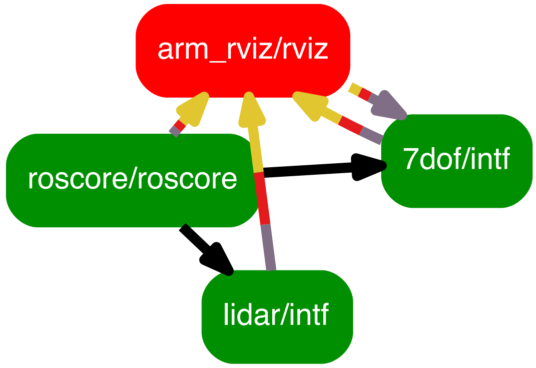


(a) Software

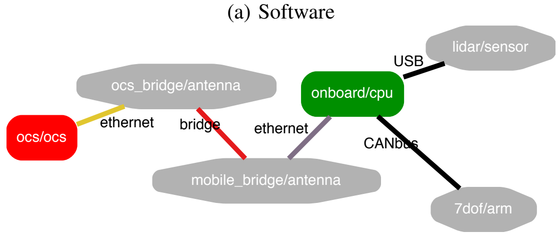


(b) Hardware

Fig. 22. Teleoperated Arm Design: Day, Indoor, Beacon Context

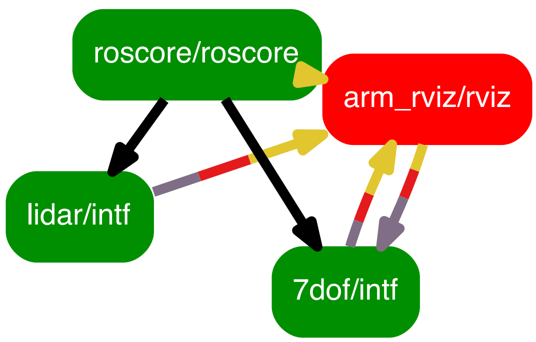


(a) Software

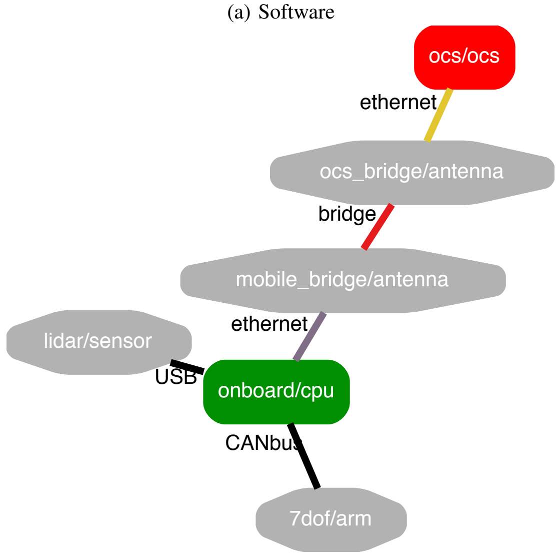


(b) Hardware

Fig. 23. Teleoperated Arm Design: Day, Outaoor, Beacon Context

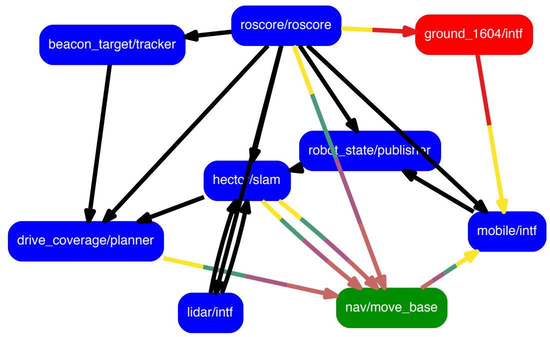


(a) Software

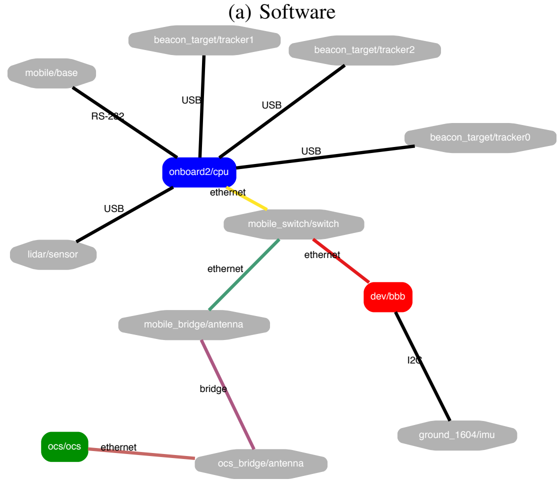


(b) Hardware

Fig. 24. Teleoperated Arm Design: Day, Outaoor, Visual Context

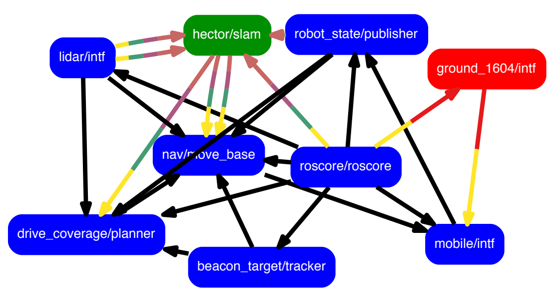


(a) Software

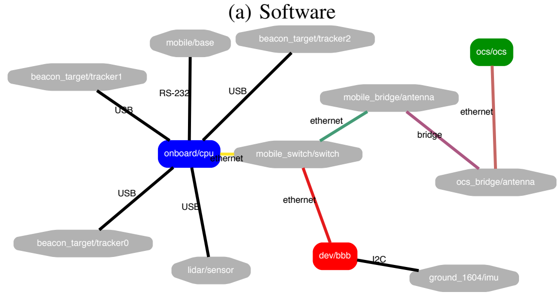


(b) Hardware

Fig. 25. Drive with Planner Design: Night, Indoor, Visual Context

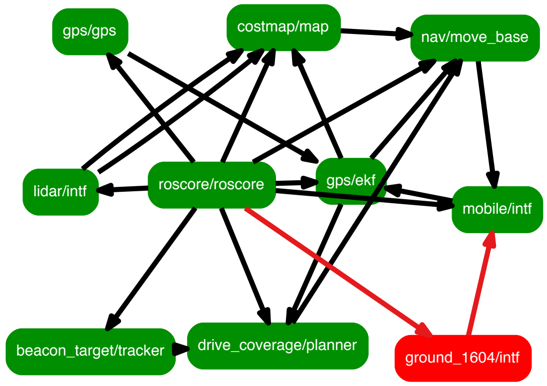


(a) Software

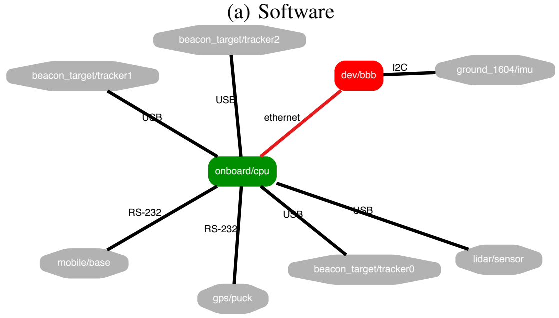


(b) Hardware

Fig. 26. Drive with Planner Design: Night, Indoor, Beacon Context

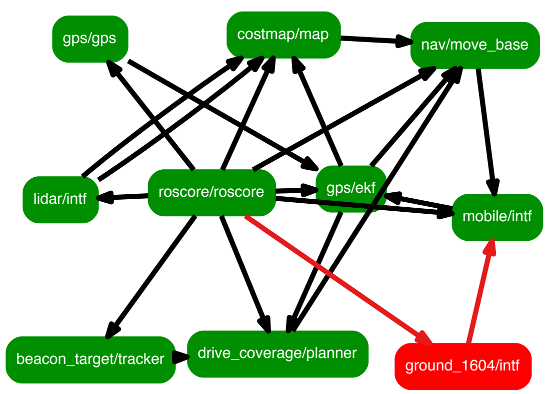


(a) Software

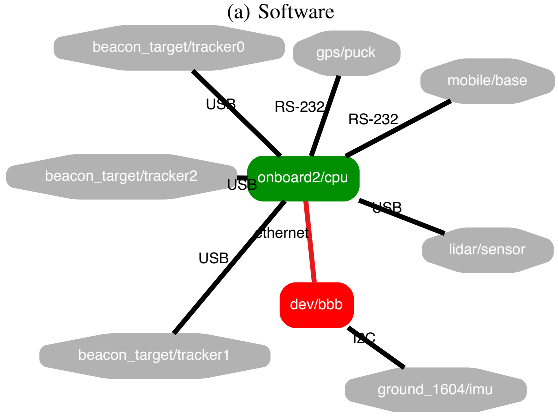


(b) Hardware

Fig. 27. Drive with Planner Design: Night, Oupdoor, Visual Context

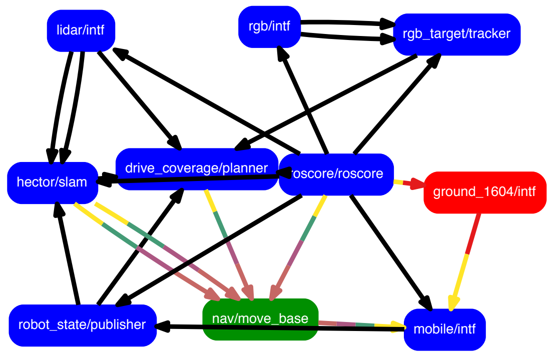


(a) Software

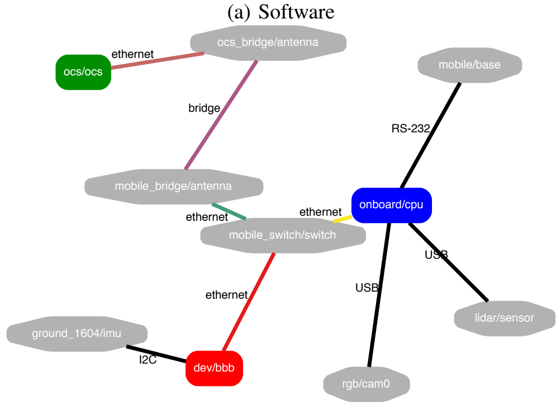


(b) Hardware

Fig. 28. Drive with Planner Design: Night, Oupdoor, Beacon Context

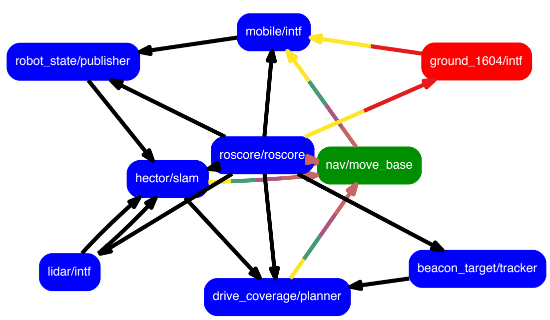


(a) Software

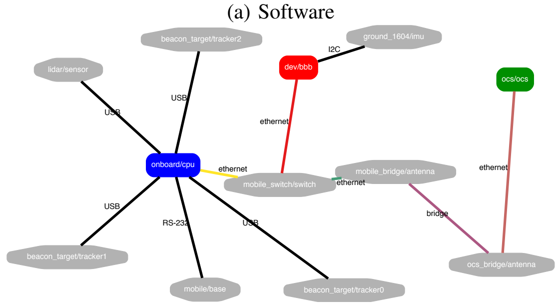


(b) Hardware

Fig. 29. Drive with Planner Design: Day, Indoor, Visual Context

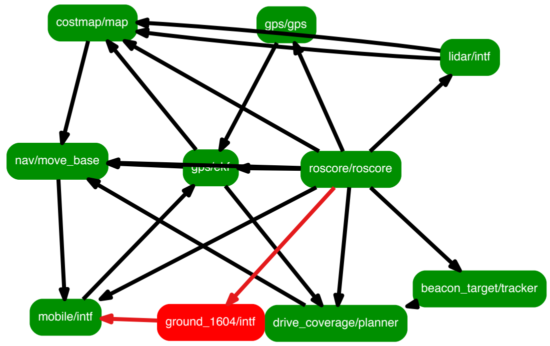


(a) Software

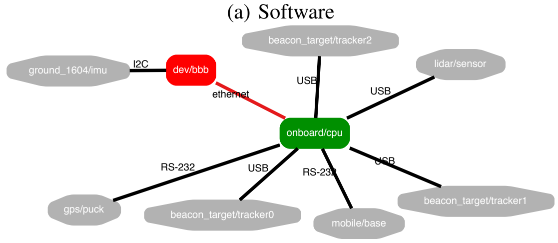


(b) Hardware

Fig. 30. Drive with Planner Design: Day, Indoor, Beacon Context

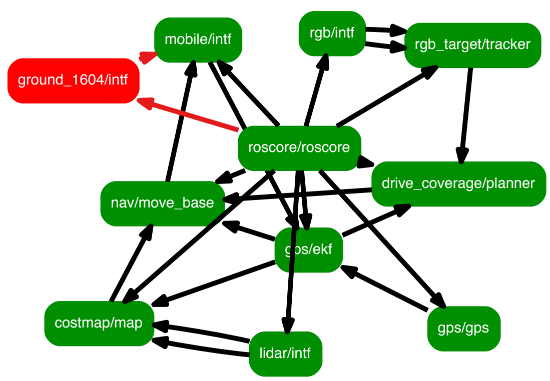


(a) Software

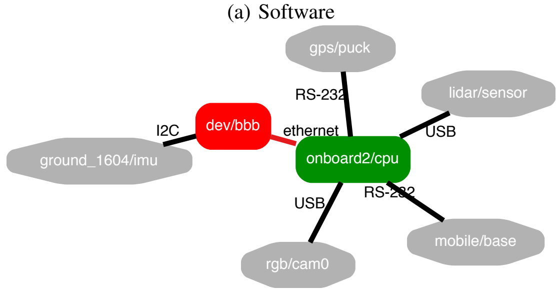


(b) Hardware

Fig. 31. Drive with Planner Design: Day, Oupdoor, Beacon Context

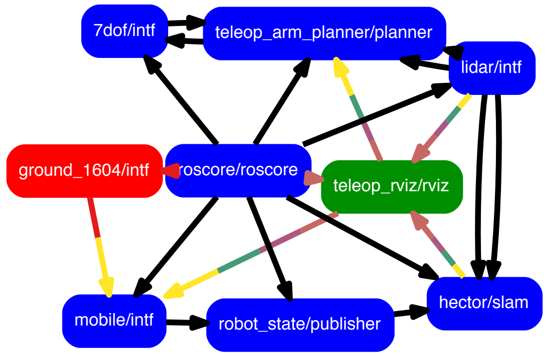


(a) Software

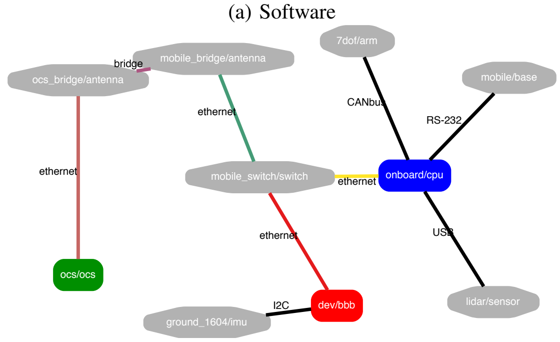


(b) Hardware

Fig. 32. Drive with Planner Design: Day, Oupdoor, Visual Context

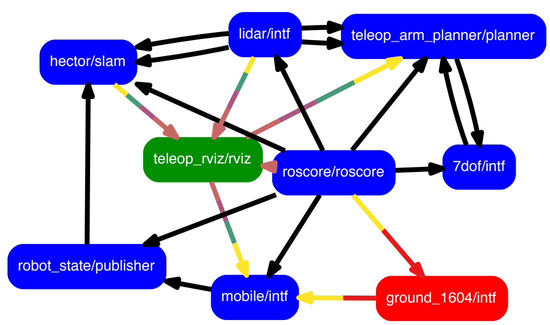


(a) Software

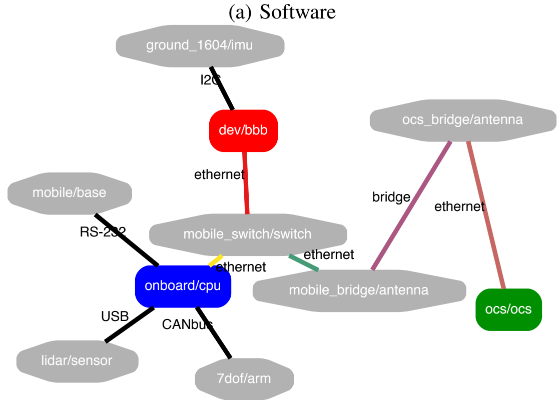


(b) Hardware

Fig. 33. Arm with Planner Design: Night, Indoor, Visual Context

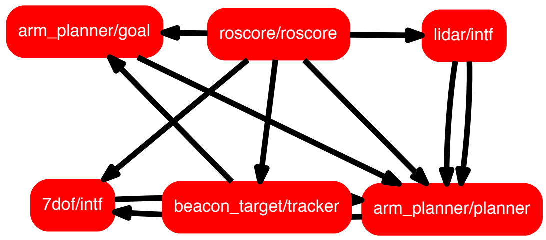


(a) Software

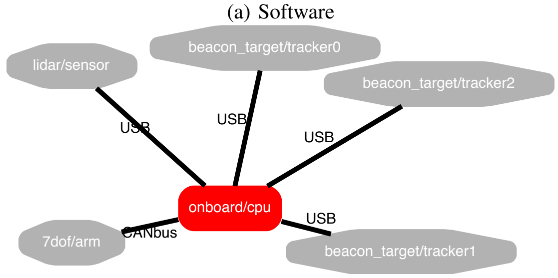


(b) Hardware

Fig. 34. Arm with Planner Design: Night, Indoor, Beacon Context

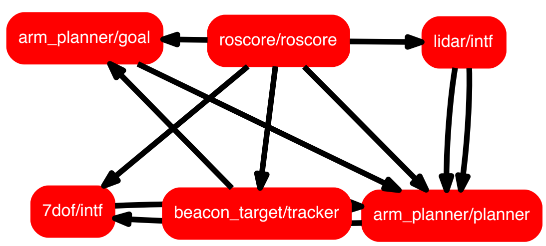


(a) Software

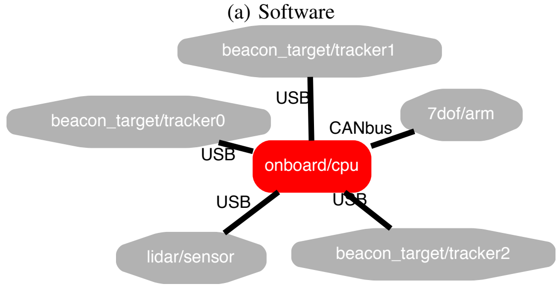


(b) Hardware

Fig. 35. Arm with Planner Design: Night, Outdoor, Visual Context

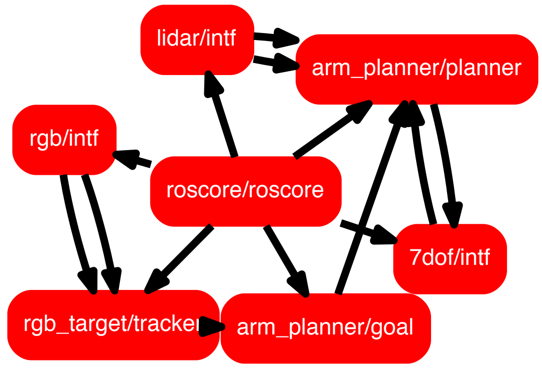


(a) Software

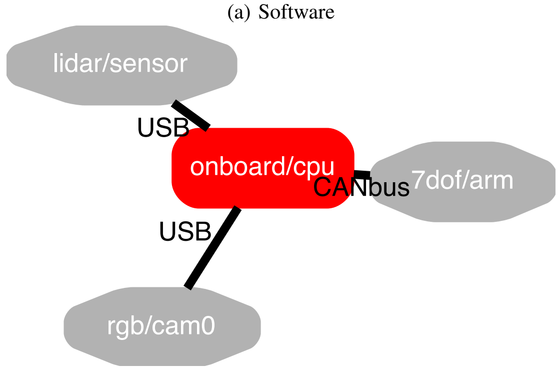


(b) Hardware

Fig. 36. Arm with Planner Design: Night, Outdoor, Beacon Context

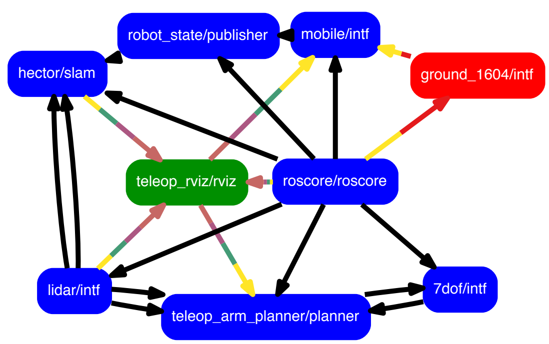


(a) Software

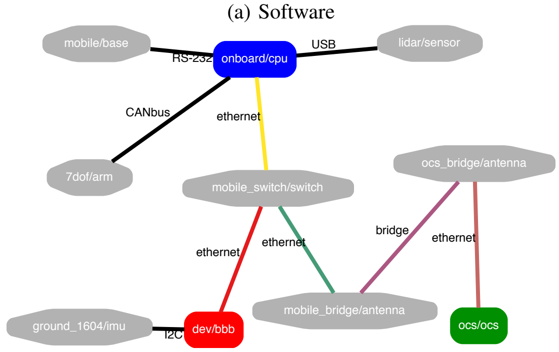


(b) Hardware

Fig. 37. Arm with Planner Design: Day, Indoor, Visual Context

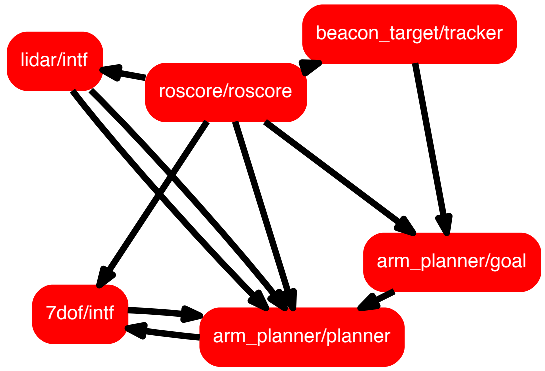


(a) Software

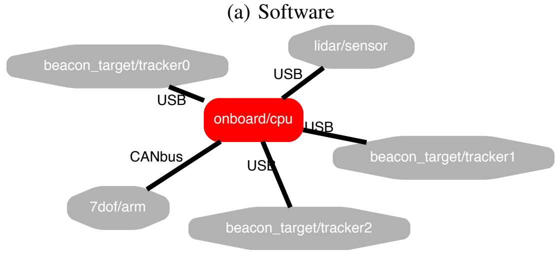


(b) Hardware

Fig. 38. Arm with Planner Design: Day, Indoor, Beacon Context

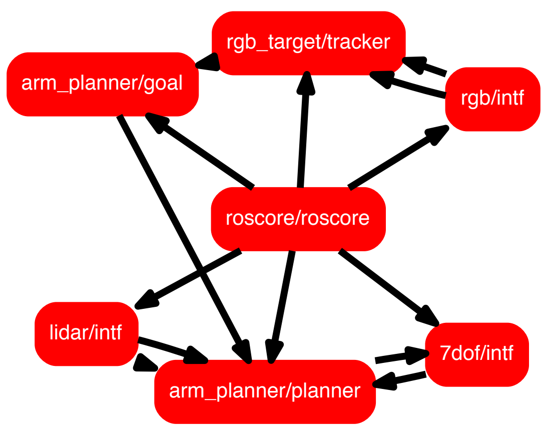


(a) Software

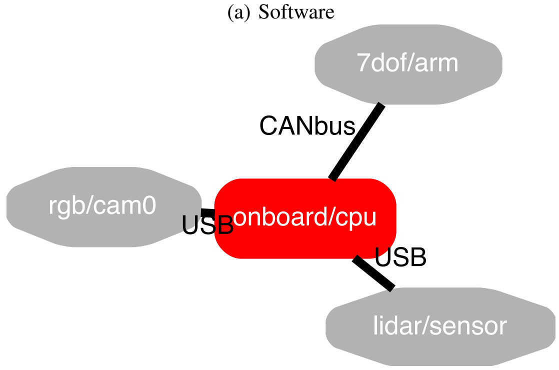


(b) Hardware

Fig. 39. Arm with Planner Design: Day, Outdoor, Beacon Context

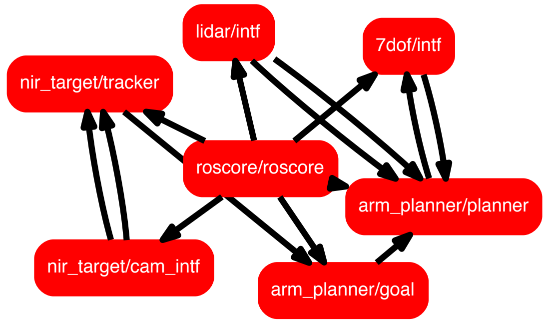


(a) Software

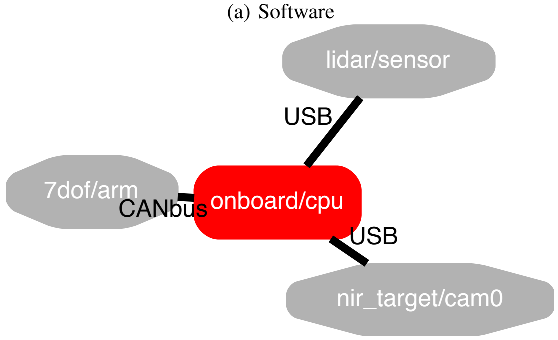


(b) Hardware

Fig. 40. Arm with Planner Design: Day, Outdoor, Visual Context

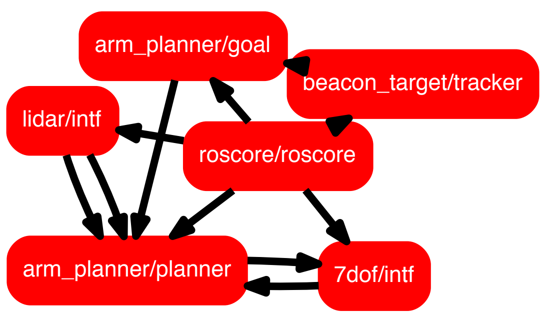


(a) Software

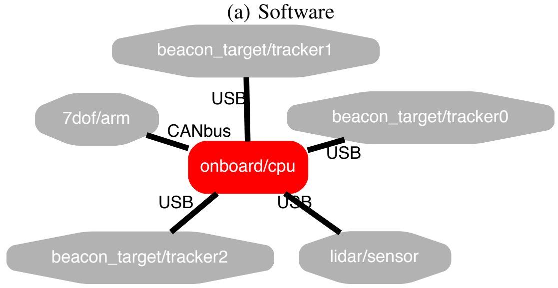


(b) Hardware

Fig. 41. Tracking, Arm Planning Design: Night, Indoor, Visual Context

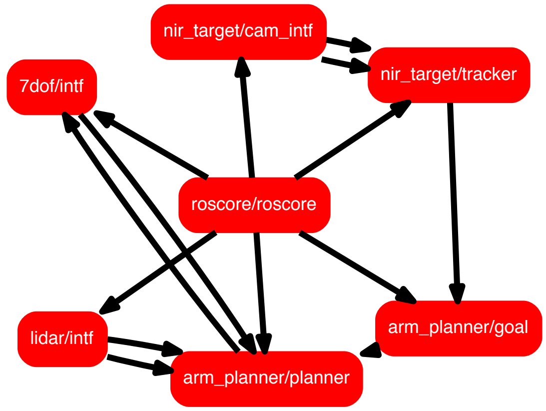


(a) Software

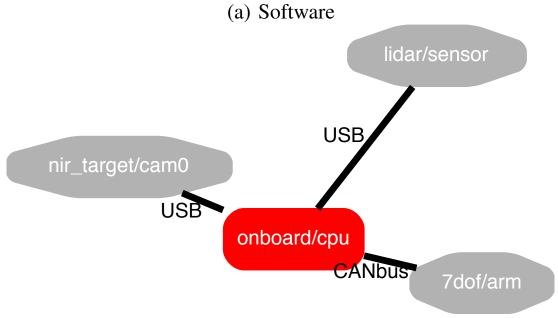


(b) Hardware

Fig. 42. Tracking, Arm Planning Design: Night, Indoor, Beacon Context

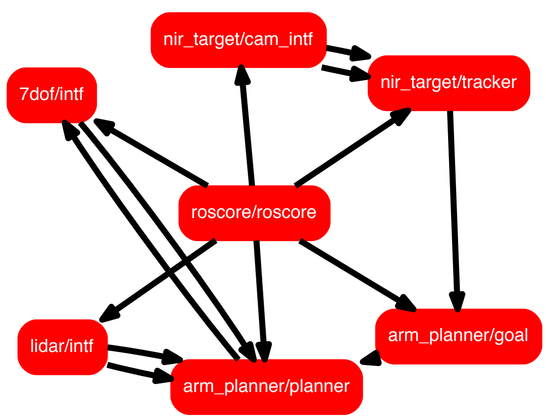


(a) Software

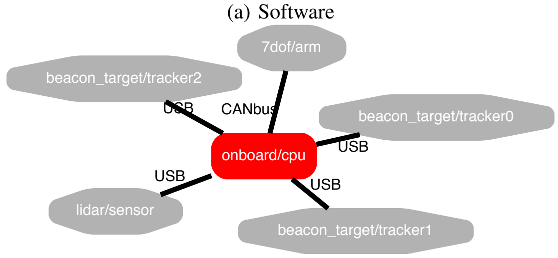


(b) Hardware

Fig. 43. Tracking, Arm Planning Design: Night, Outdoor, Visual Context

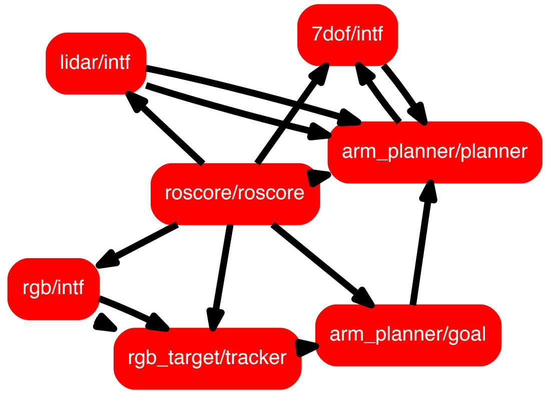


(a) Software

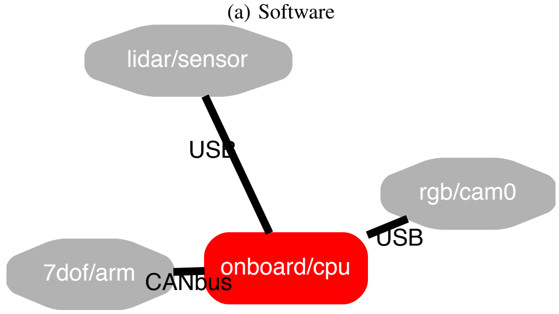


(b) Hardware

Fig. 44. Tracking, Arm Planning Design: Night, Outdoor, Beacon Context

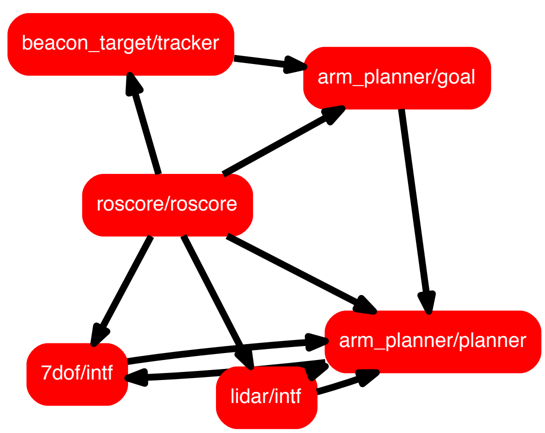


(a) Software

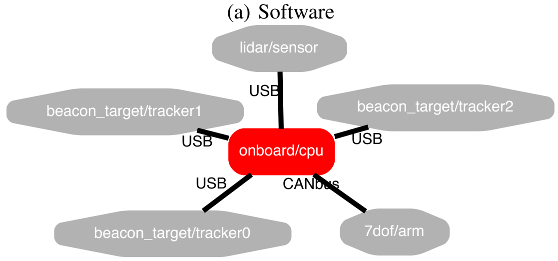


(b) Hardware

Fig. 45. Tracking, Arm Planning Design: Day, Indoor, Visual Context

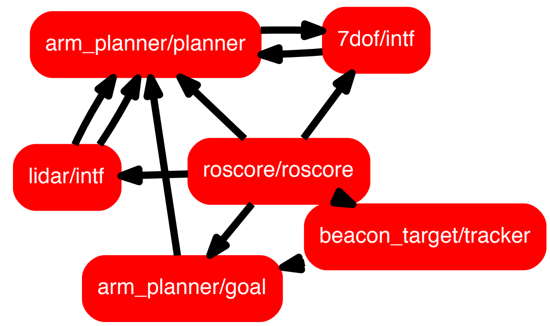


(a) Software

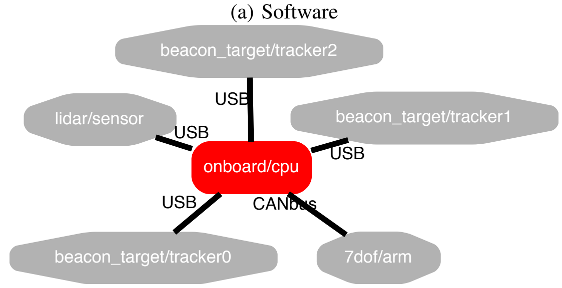


(b) Hardware

Fig. 46. Tracking, Arm Planning Design: Day, Indoor, Beacon Context

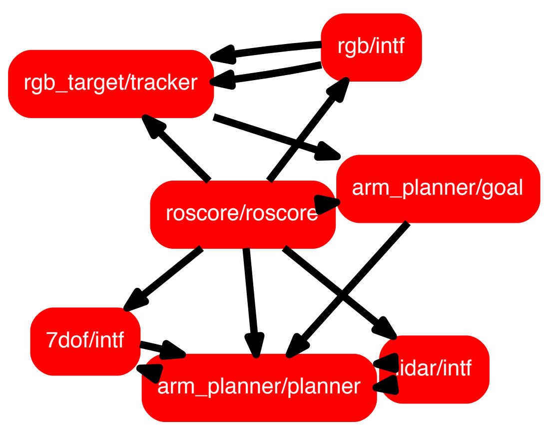


(a) Software

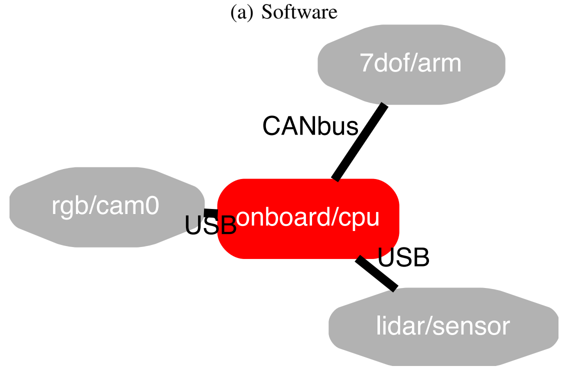


(b) Hardware

Fig. 47. Tracking, Arm Planning Design: Day, Outdoor, Beacon Context

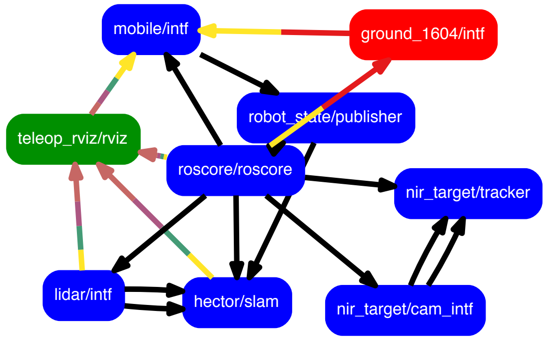


(a) Software

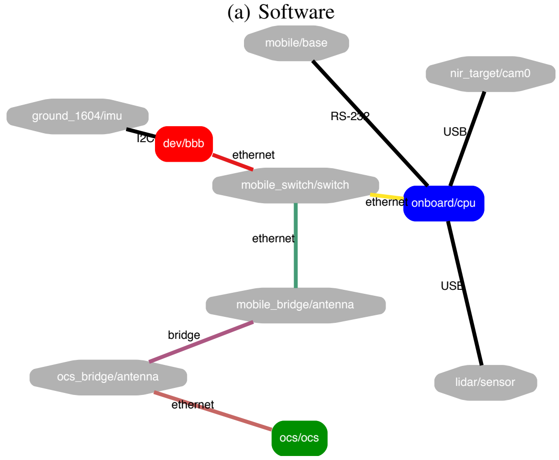


(b) Hardware

Fig. 48. Tracking, Arm Planning Design: Day, Outdoor, Visual Context

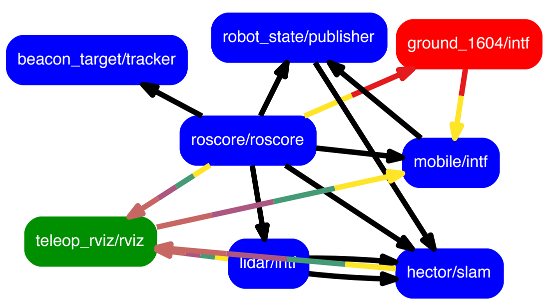


(a) Software

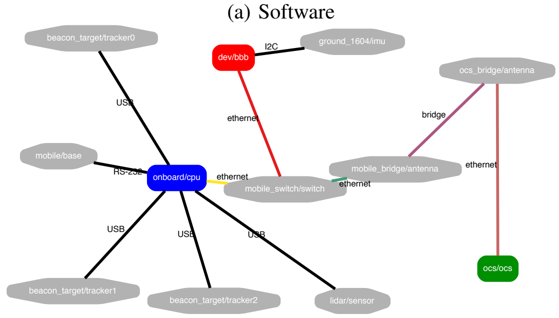


(b) Hardware

Fig. 49. Tracking, Driving Design: Night, Indoor, Visual Context

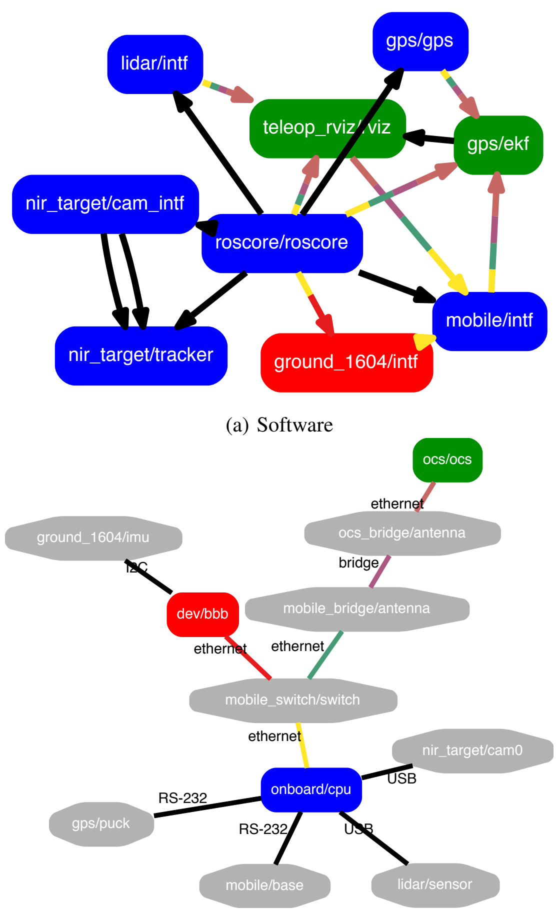


(a) Software



(b) Hardware

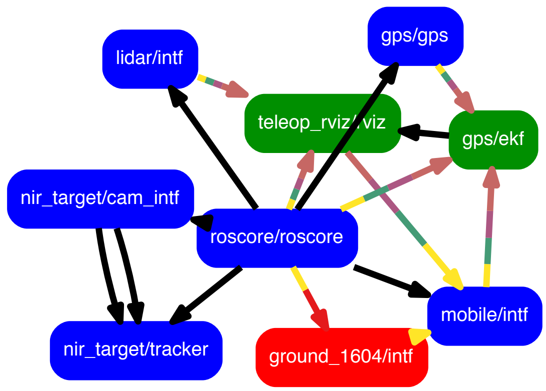
Fig. 50. Tracking, Driving Design: Night, Indoor, Beacon Context



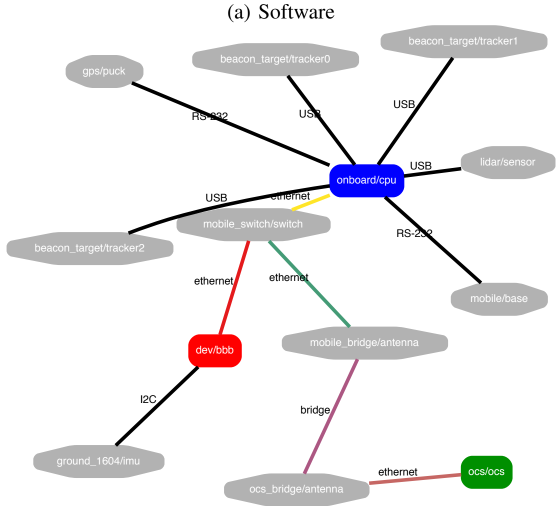
(a) Software

(b) Hardware

Fig. 51. Tracking, Driving Design: Night, Outdoor, Visual Context

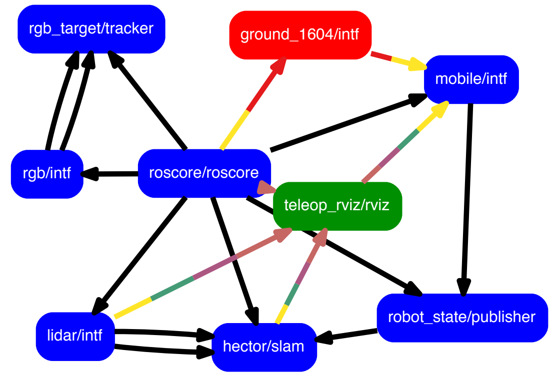


(a) Software

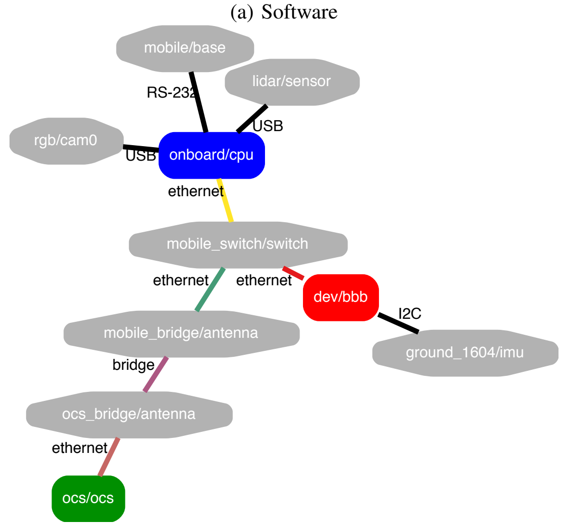


(b) Hardware

Fig. 52. Tracking, Driving Design: Night, Outdoor, Beacon Context

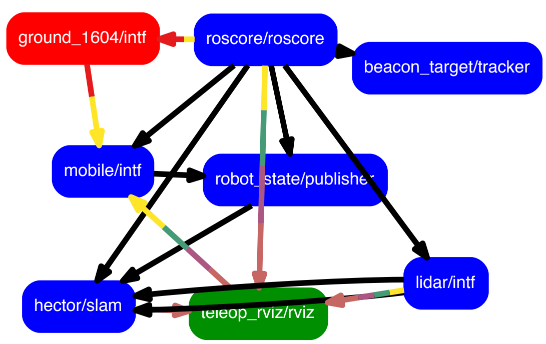


(a) Software

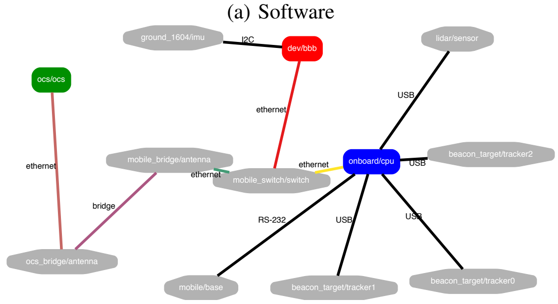


(b) Hardware

Fig. 53. Tracking, Driving Design: Day, Indoor, Visual Context

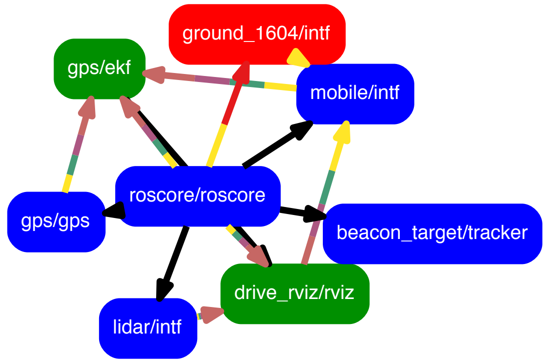


(a) Software

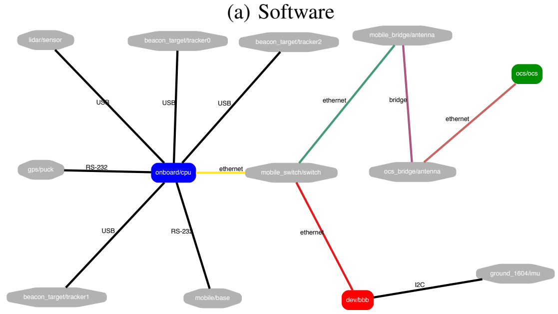


(b) Hardware

Fig. 54. Tracking, Driving Design: Day, Indoor, Beacon Context

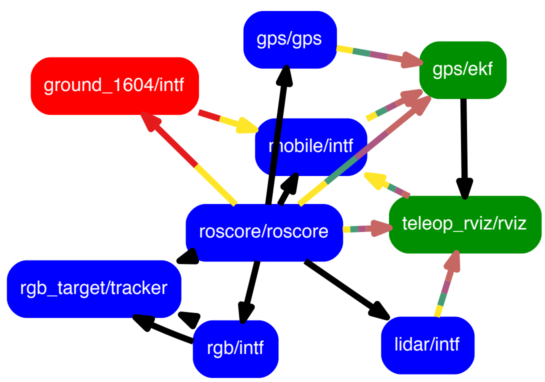


(a) Software

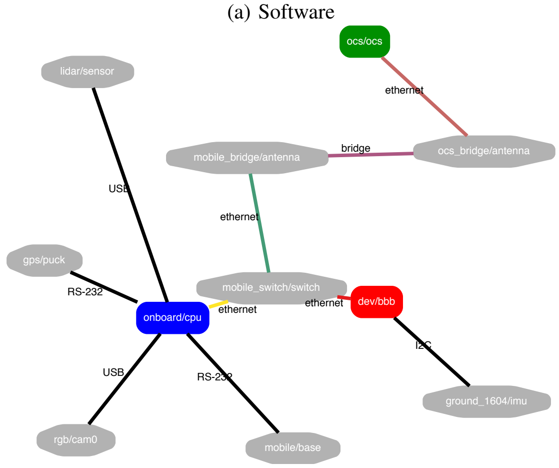


(b) Hardware

Fig. 55. Tracking, Driving Design: Day, Outdoor, Beacon Context

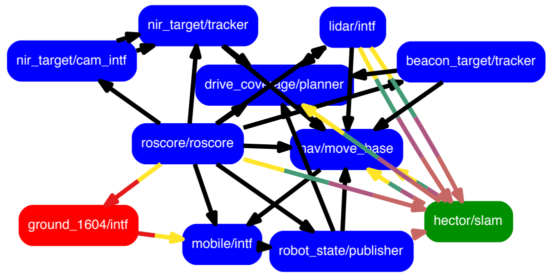


(a) Software

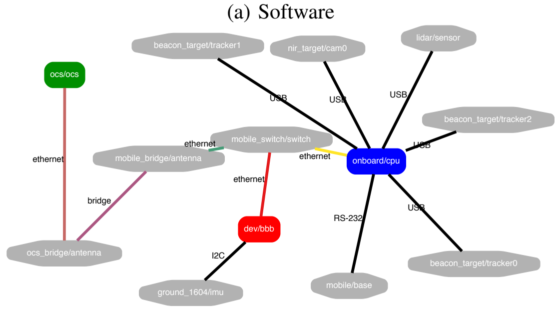


(b) Hardware

Fig. 56. Tracking, Driving Design: Day, Outdoor, Visual Context

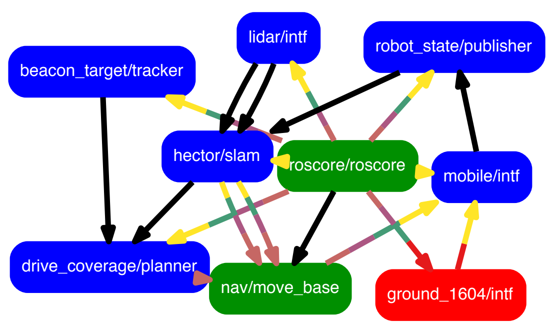


(a) Software

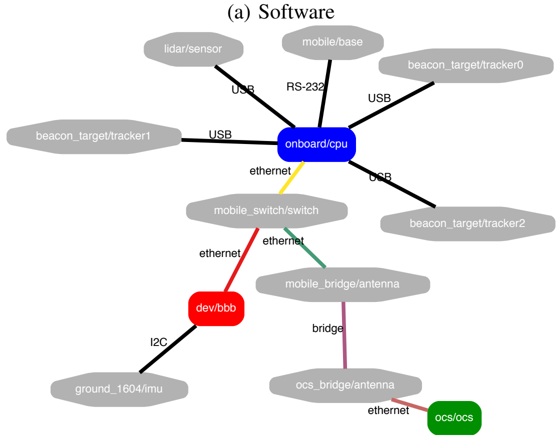


(b) Hardware

Fig. 57. Tracking, Driving, Planning Design: Night, Indoor, Visual Context

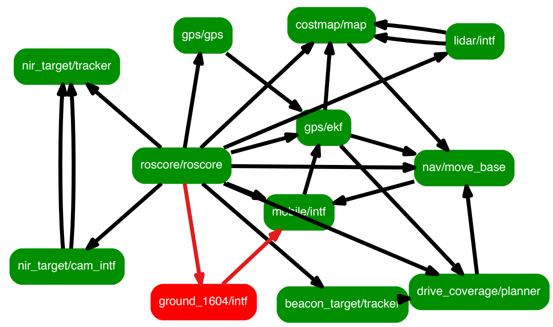


(a) Software

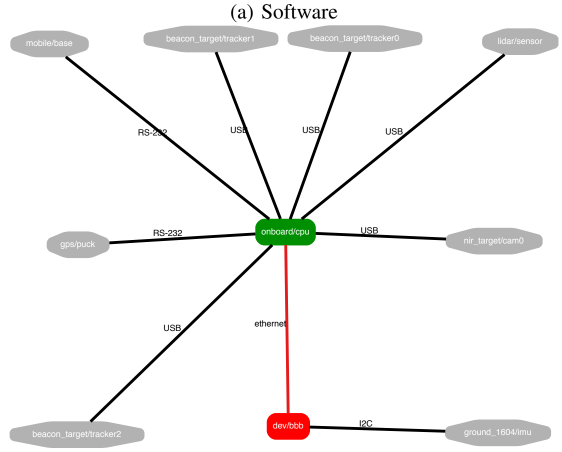


(b) Hardware

Fig. 58. Tracking, Driving, Planning Design: Night, Indoor, Beacon Context

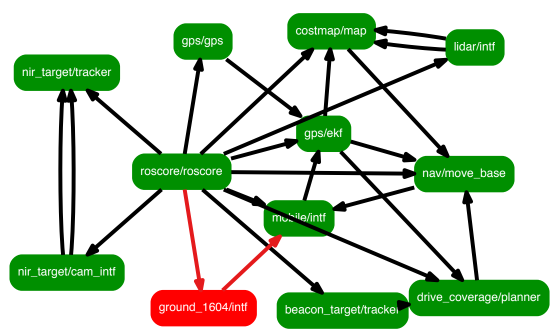


(a) Software

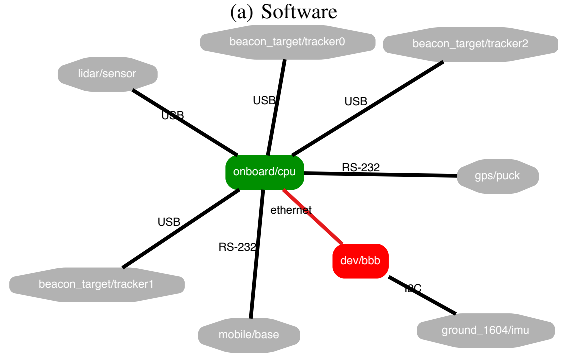


(b) Hardware

Fig. 59. Tracking, Driving, Planning Design: Night, Outdoor, Visual Context

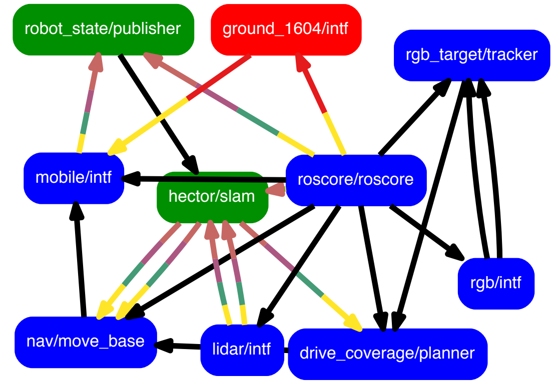


(a) Software

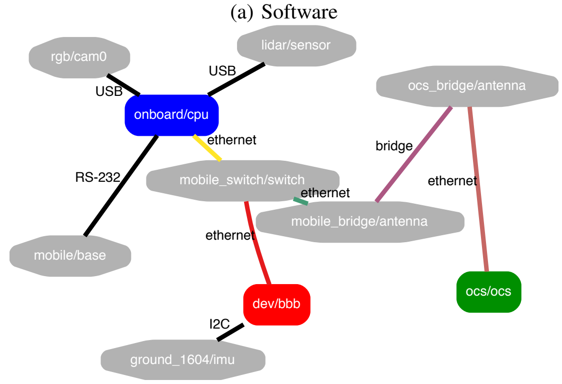


(b) Hardware

Fig. 60. Tracking, Driving, Planning Design: Night, Outdoor, Beacon Context

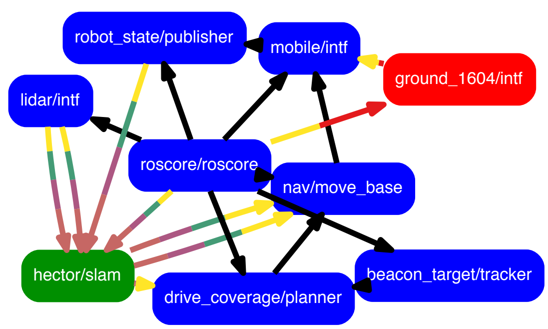


(a) Software

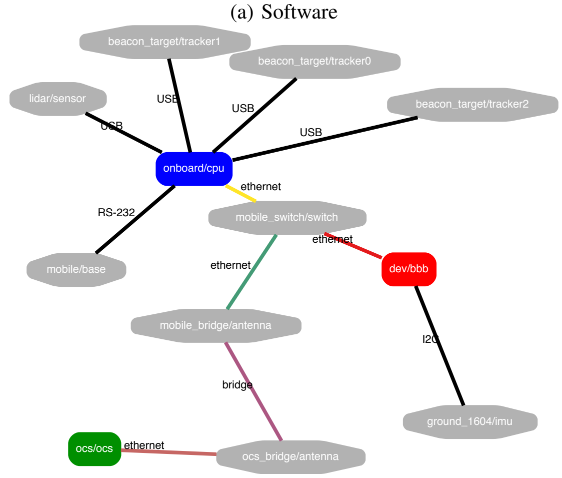


(b) Hardware

Fig. 61. Tracking, Driving, Planning Design: Day, Indoor, Visual Context

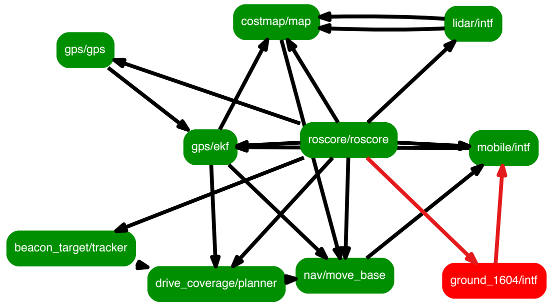


(a) Software

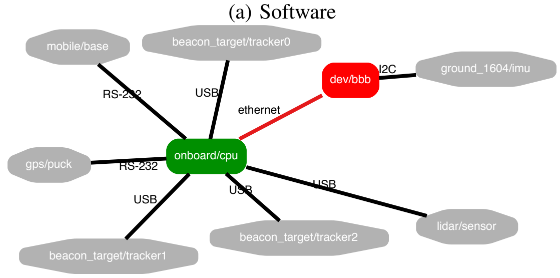


(b) Hardware

Fig. 62. Tracking, Driving, Planning Design: Day, Indoor, Beacon Context

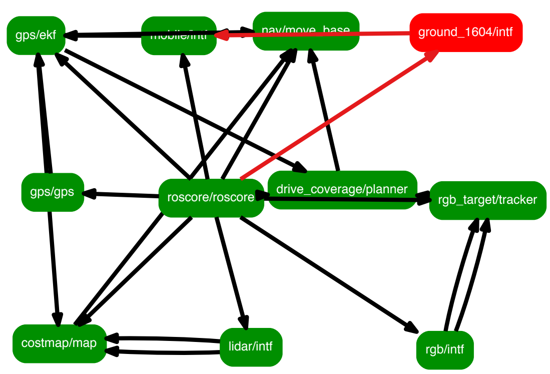


(a) Software

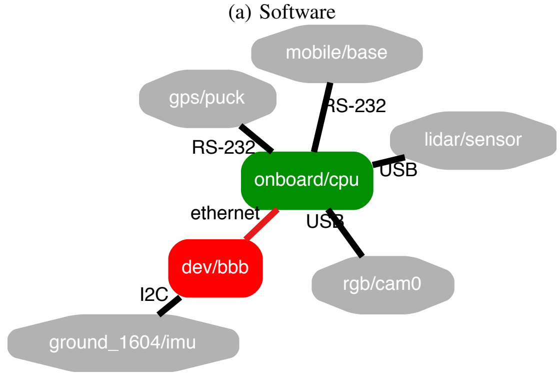


(b) Hardware

Fig. 63. Tracking, Driving, Planning Design: Day, Outdoor, Beacon Context

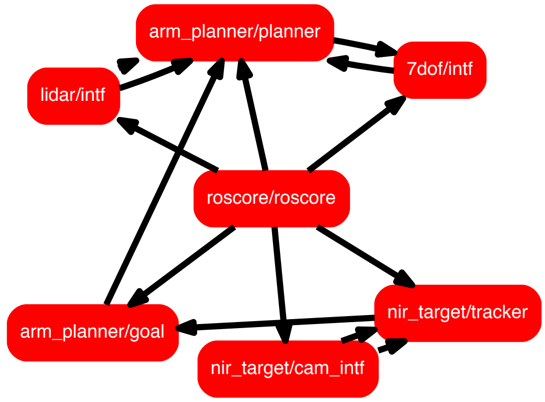


(a) Software

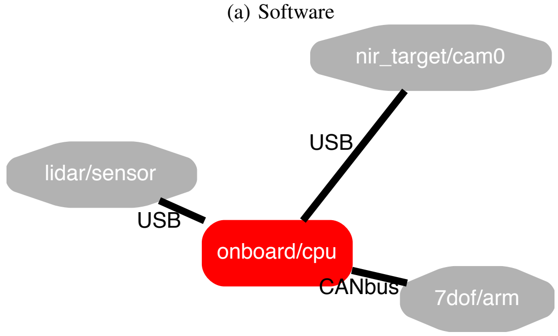


(b) Hardware

Fig. 64. Tracking, Driving, Planning Design: Day, Outdoor, Visual Context

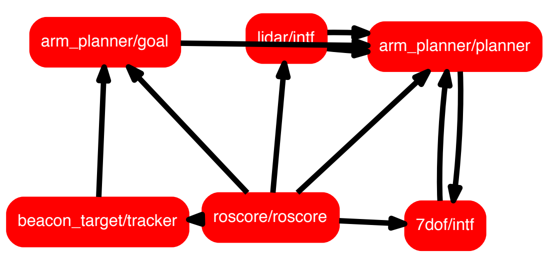


(a) Software

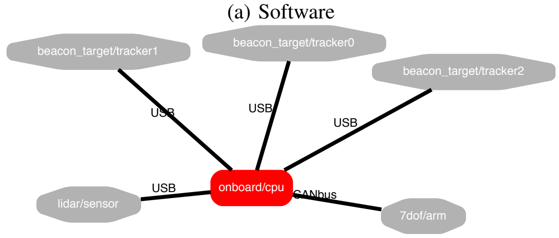


(b) Hardware

Fig. 65. Tracking, Driving, Arm Design: Night, Indoor, Visual Context

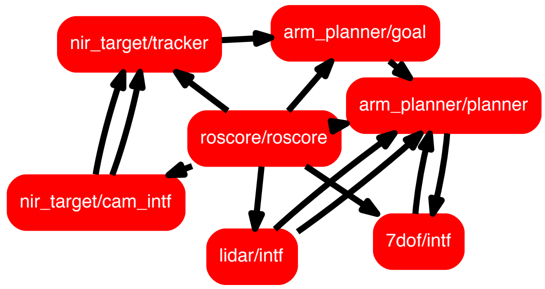


(a) Software

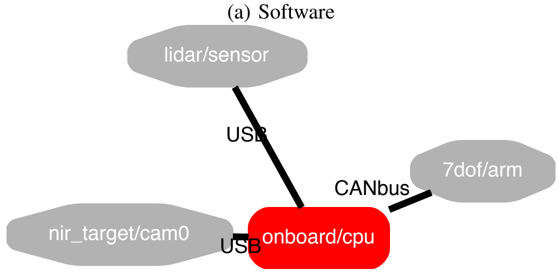


(b) Hardware

Fig. 66. Tracking, Driving, Arm Design: Night, Indoor, Beacon Context

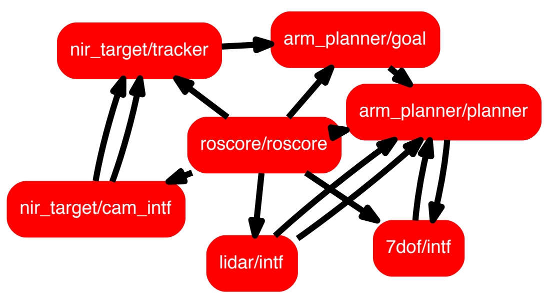


(a) Software

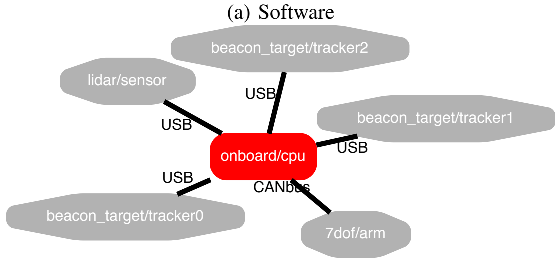


(b) Hardware

Fig. 67. Tracking, Driving, Arm Design: Night, Outdoor, Visual Context

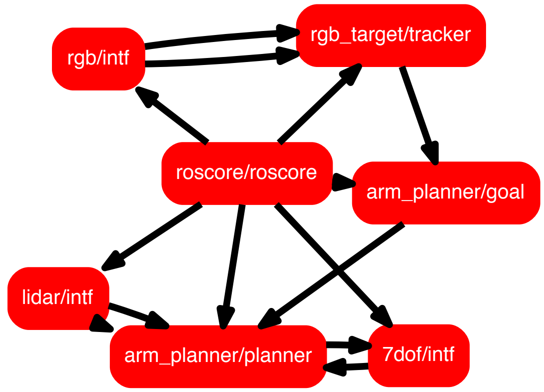


(a) Software

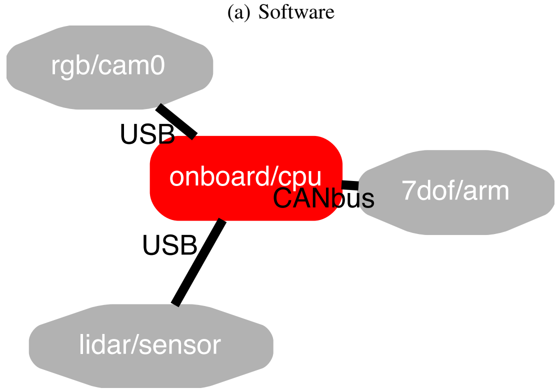


(b) Hardware

Fig. 68. Tracking, Driving, Arm Design: Night, Outdoor, Beacon Context

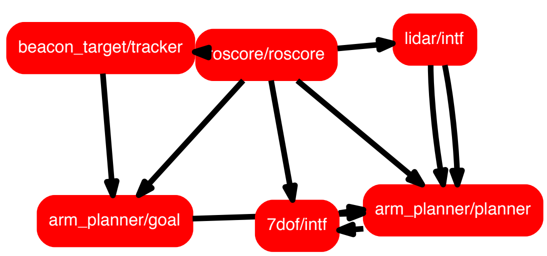


(a) Software

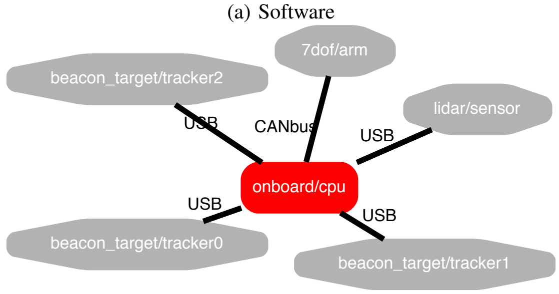


(b) Hardware

Fig. 69. Tracking, Driving, Arm Design: Day, Indoor, Visual Context

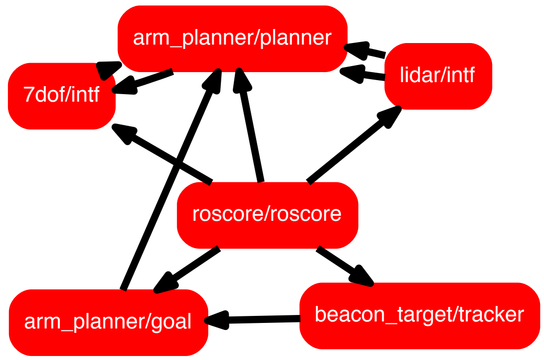


(a) Software

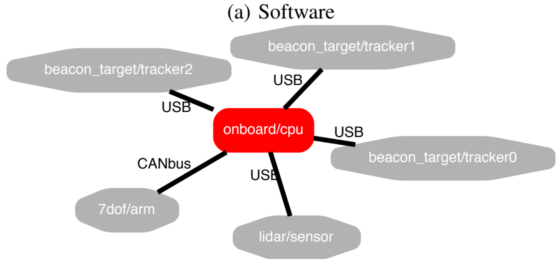


(b) Hardware

Fig. 70. Tracking, Driving, Arm Design: Day, Indoor, Beacon Context

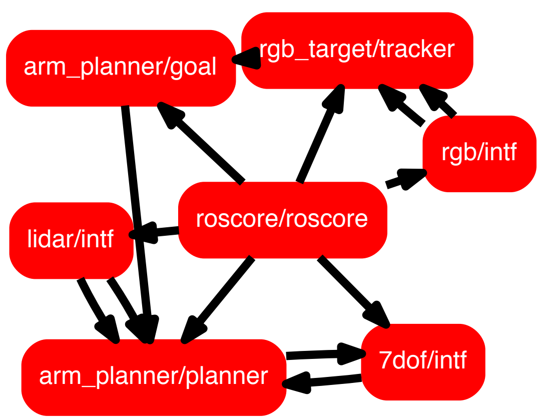


(a) Software

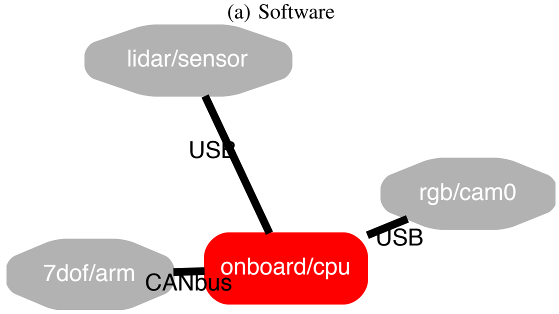


(b) Hardware

Fig. 71. Tracking, Driving, Arm Design: Day, Outdoor, Beacon Context

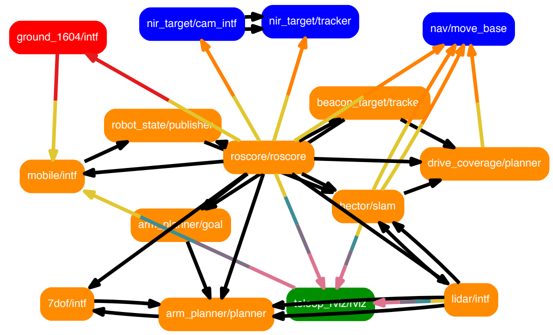


(a) Software

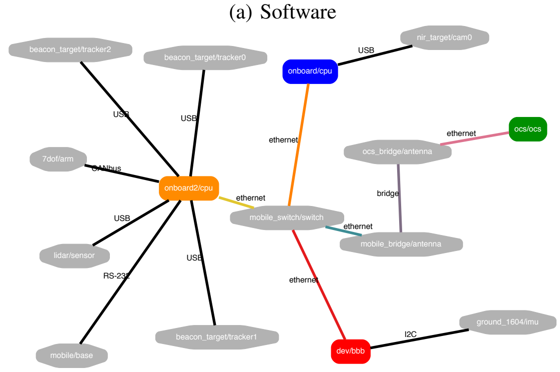


(b) Hardware

Fig. 72. Tracking, Driving, Arm Design: Day, Outdoor, Visual Context

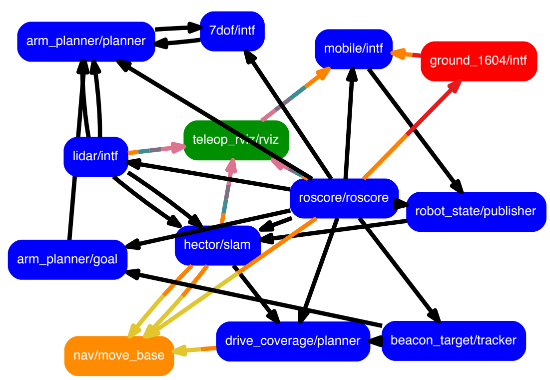


(a) Software

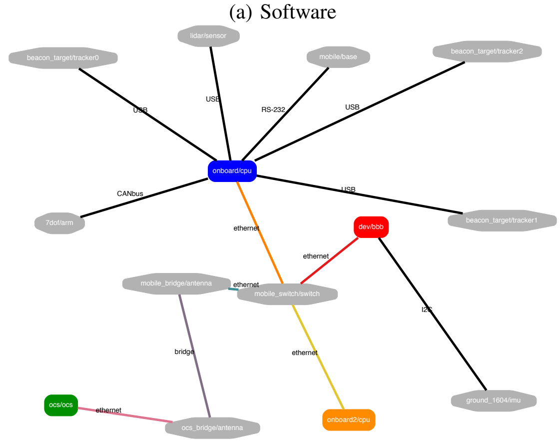


(b) Hardware

Fig. 73. Full Functionality Design: Night, Indoor, Visual Context

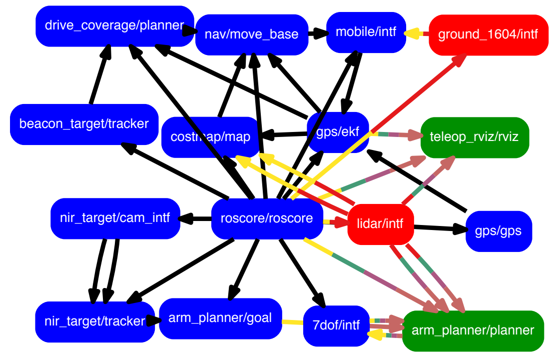


(a) Software

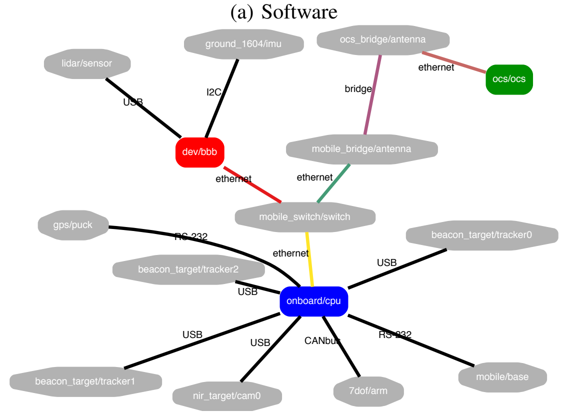


(b) Hardware

Fig. 74. Full Functionality Design: Night, Indoor, Beacon Context

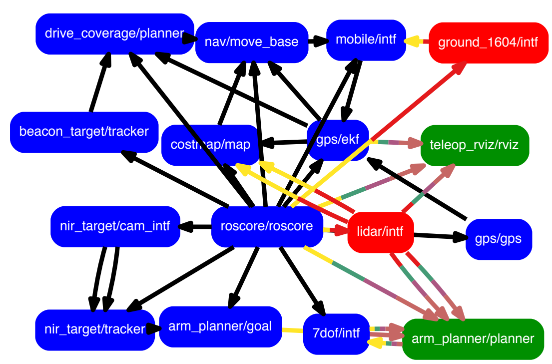


(a) Software

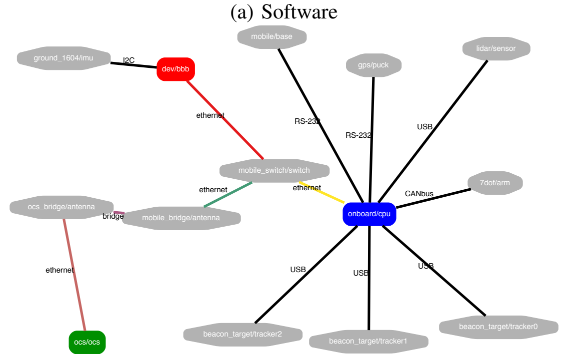


(b) Hardware

Fig. 75. Full Functionality Design: Night, Outdoor, Visual Context

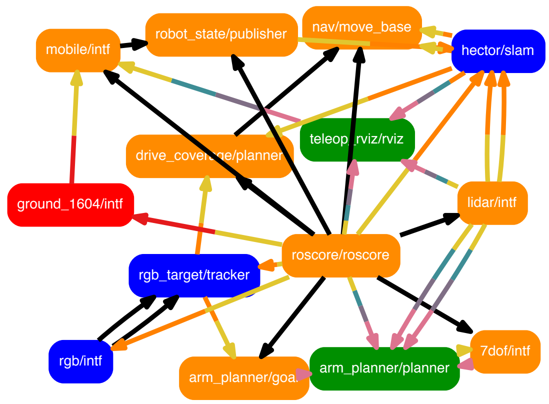


(a) Software

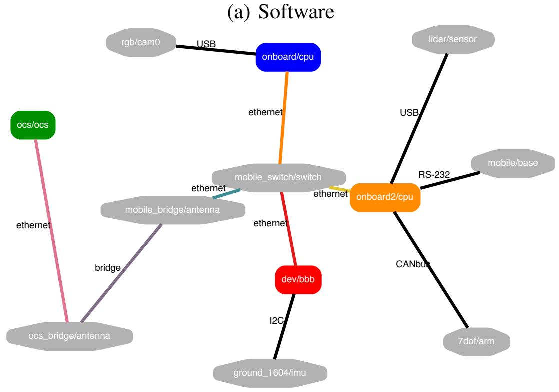


(b) Hardware

Fig. 76. Full Functionality Design: Night, Outdoor, Beacon Context

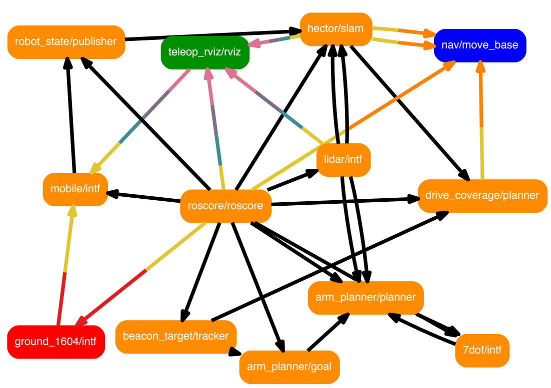


(a) Software

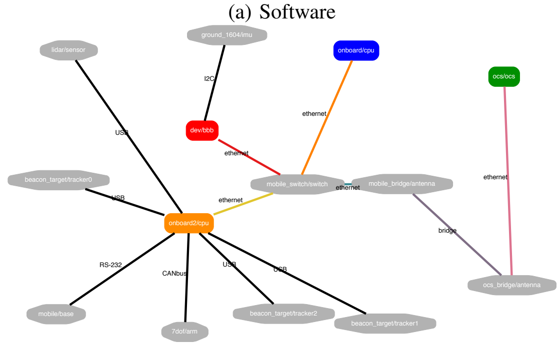


(b) Hardware

Fig. 77. Full Functionality Design: Day, Indoor, Visual Context

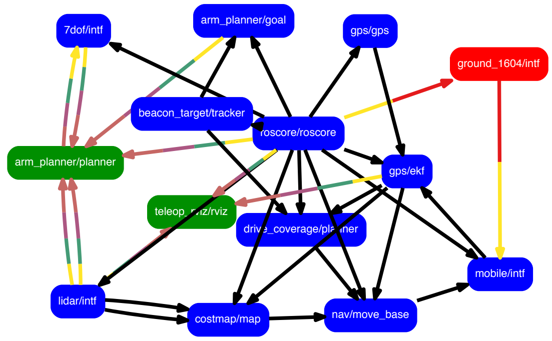


(a) Software

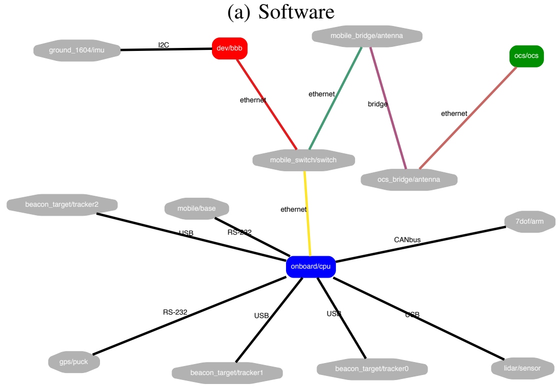


(b) Hardware

Fig. 78. Full Functionality Design: Day, Indoor, Beacon Context

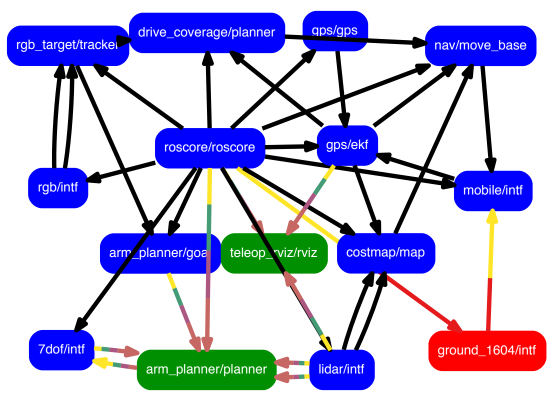


(a) Software

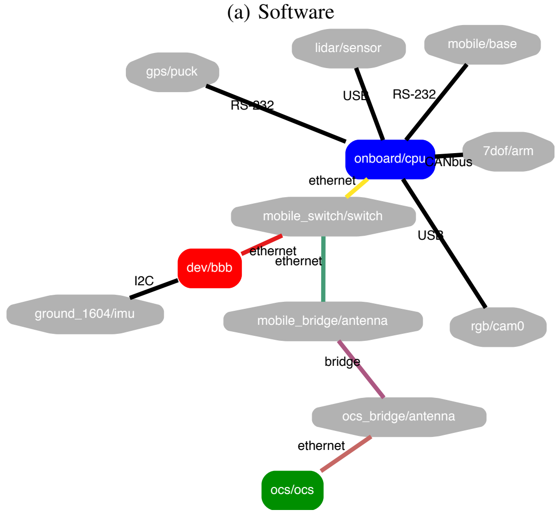


(b) Hardware

Fig. 79. Full Functionality Design: Day, Outdoor, Beacon Context



(a) Software



(b) Hardware

Fig. 80. Full Functionality Design: Day, Outdoor, Visual Context