6.111 Final Project Proposal
Kelly Snyder and Rebecca Greene

Abstract

The Cambot project proposes to build a robot using two distinct FPGAs that will interact with users wirelessly, using the labkit, a vga screen, and an ceiling-mounted camera as an interface. Our base goal is to allow the user to control the robot using buttons on the labkit, while the camera tracks it on the screen. Once this has been accomplished, boundary detection and mapping algorithms will be added to the system. The user would be able to select if they wish to control the hexapod manually, or guide it to specific points using the screen. When given a point, the labkit will generate a set of directions to move the robot towards a desired location. A third goal would be to have the labkit recognize two different images- that of the hexapod and that of a marker, and it would generate algorithms to make the robot chase the marker object.

The robot itself will be a hexapod, consisting of six legs attached to a circular base, controlled by a nexys3 fpga that will be mounted to it. Each leg will have two small servo motors attached to it, and these twelve motors can be used to propel the robot in any direction within a two dimensional plane. Instructions will be transmitted to the hexapod wirelessly, using the XBee 802.15.4 RF module. Proximity sensors on the robot will prevent it from crashing into any objects immediately in it's path.

Block Diagram

As can be seen in Figure 1 (below), our project is split into two main sections, each with its own FPGA. One section will control the hexapod itself, while the other section will take input from the user and from the ceiling mounted-camera and turn that input into instructions for the hexapod.

Inputs to the system will consist of data from the camera and user input on the labkit (buttons, switches, etc.) The user can control the hexapod's movement directly using buttons, or the hexapod can move autonomously to a point specified by the user. Button input is enabled with a switch, and in this mode debounced signals from the buttons would be passed through the Main Labkit FSM to the Xbee module, and transmitted to the hexapod. In the alternative case, data from the camera would feed into the Image Recognition Module and the Screen Module. The latter would display the robot's location, and interactions between the Screen module and the Labit Main FSM would produce coordinates for the robot to travel towards. Using coordinates from the Main FSM and the Image Recognition Module, the Mapping Module would produce directions to pass through the Main Labkit FSM to the Xbee.

On the other side of the system, the Xbee module on the hexapod receives data from the labkit, and passes it to the hexapod's Main FSM. This will use data from the proximity sensor module to decide if it's appropriate to move, and selectively send the directions to the Motor FSM to eventually be translated into motion.

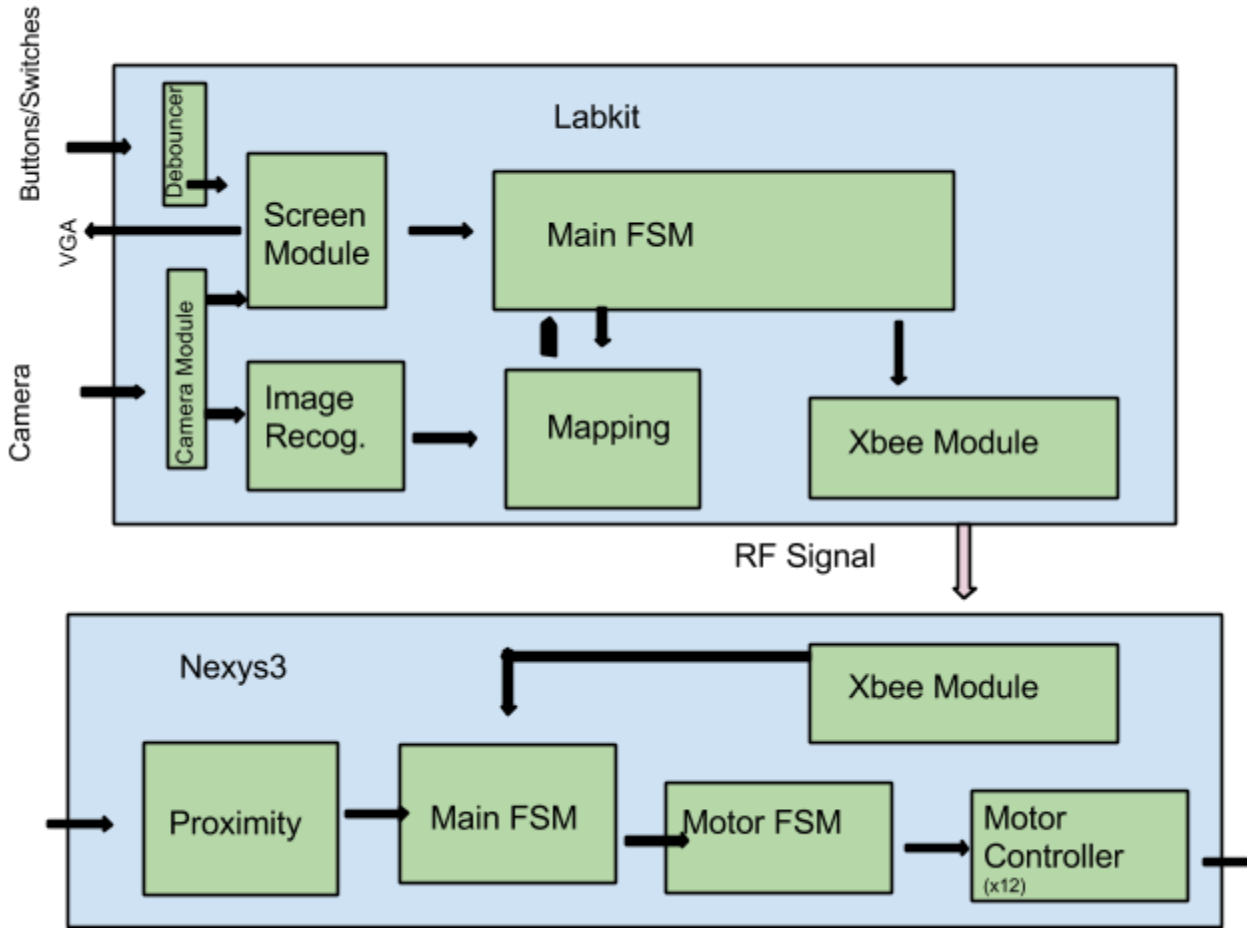

Figure 1. This diagram shows the basic high level flow of our system. It can be seen that the system is divided into two main parts, which then are divided into smaller parts.

Modules

The high level modules which can be seen in Figure 1 (shown in green) will be split evenly between Kelly and Becca. In general, Becca will be working with the Nexys 3 and the hexapod, and Kelly will be working with the labkit and video. We have not yet determined who will work with the distance sensors, it will likely depend on which person's basic tasks are finished more quickly. A few sections that involve interfacing the two sections together will be worked on by both to ensure smooth integration. Again, these modules are high level and each one might actually consist of one or more verilog modules.

Main FSM- hexapod - Both

This module is the top level on the hexapod, which brings all other modules on the hexapod together. The main FSM takes input from the XBee Interface module and the Proximity Sensor Module. These both determine what information should be passed to the Motor FSM,

including direction and type of motion. For our base set of goals, the hexapod will be directed to stop if an object is immediately in it's path. As the project becomes more complicated, we will change the algorithm to make the hexapod walk around objects undetected by the camera.

Motor FSM - Becca

The Motor FSM will manage motion for the entire robot. It will receive an input from the Main FSM specifying what type of motion is necessary, at what speed, and what style. The number of options offered for motion control will vary for the first, second, and third level goals for the project. With this input, the Motor FSM will use a combination of vector algorithms and lookup tables to generate a series of turns for each motor. Using an embedded timer module, the FSM will send appropriate signals to the Motor Controllers when they need to turn.

Motor Controller - Becca

There will be twelve Motor Controller modules, and each module will be directly responsible for the motion of one motor, under the direction of the Motor FSM.. Servo motors are operated using a pwm signal, and the motion of the motor is controlled by varying the duty cycle of the pwm. This module will receive input from the motor FSM on what type of movement (if any) is necessary at a given point in time, and will generate the appropriate pwm output signal to the motor. Some additional analog circuitry will be necessary to convert the output of the fpga to the power levels supplied by the motor.

XBee Interface - Becca

The XBee Interface module will acquire and transmit data using the Xbee RF transceiver. The Xbee communicates using serial protocols, although it can also be programmed with analog signals. A lofty goal would be to implement serial communication protocols within this module, although we will probably start by using the fpga to trigger analog inputs to the Xbee, causing it to enter different pre-programmed states. The module must both translate information into a form that can be transmitted, and then translate transmitted data back into a form that is useful to the state machines of the fpgas. Two such modules are necessary- one within the labkit and one within the hexapod. Transmitted data will contain information on what direction the robot will be traveling in, and potentially also information about what type of motion the robot should be using to ambulate.

Proximity Sensor - Becca

The Proximity Sensor module will take signals from distance sensors on the robot and translate them into information that is useful to the Main FSM. These will be used to ensure that the robot doesn't crash into anything. Output from the sensors will be passed through A/D converters before reaching the input pins of the Nexys3. Combining data from all sensors, the fpga will determine whether or not it is walking towards an object. If so, iIt will send a warning flag to the Main FSM, as well a second output describing the approximate location of that object.

Main FSM- labkit - Both

This module, like the main FSM on the hexapod, is essentially the top level which brings together the various modules on the labkit. It also transmits via Xbee the various signals it processes for the hexapod's motion. For the basic level this will consist of inputs from various buttons, and in higher level implementation will involve directions based on image processing and mapping. A switch will control whether the user is controlling the robot, or if it's being guided by the mapping module. All of this will be brought together and transmitted in this module.

Camera Interface - Kelly

This module will interface the camera with the rest of the system. The camera will be connected to the lab kit via an RCA connection. This module will take the data from the camera as an input and convert it to digital video data, which can be used and further processed by the main system. The digital video data which is outputted will be used as an input by both the Screen Interface module, to be displayed on a monitor, and, later, the image recognition module, which will analyze the data.

Screen Interface - Kelly

This module will take the digital video data from the camera module and process it to be displayed on a monitor. More specifically, it takes as an input the video data and processes it to determine various necessary timing and color parameters (hsync, vsync, RGB_out, etc). Those parameters will then be used to display the video data received from the camera. In the simplest implementation of our project, the video will simply be displayed on the monitor for user reference and will have no actual impact on the movement of the hexapod. In later implementations, the user will be able to click on a part of the screen and expect the Hexapod to move to that point. In that case, this module will also interface between the user's input to the image and the main system, which will use that data to determine the hexapod's movement.

Labkit Interface/Debounce - Kelly

This simple module will debounce any buttons or other hardware on the labkit that will be used. This section will also determine what those buttons do. Currently, our basic implementation will use the up, down, left, and right push buttons to control the movement of the hexapod. (In later implementations it will move autonomously towards a specified point.) Later implementation might also include more complex, or simply more, commands from the labkit hardware to the robot, such as using the switches to control speed of motion.

Image Recognition - Kelly

This module will be used primarily in later/more advanced implementations of the design. This module will take digital video data from the Camera Interface module and use image recognition to determine various things, depending on which level of the project we are working on. For our second level goal, for instance, we will determine where the hexapod is and where

the point is that it is trying to move to. We will then use visual feedback to determine if the hexapod is moving in the right direction, correct movement if necessary and determine when it has reached the point. For our third level goal, we will determine both the location of the hexapod and the location of a second object and track the hexapod's progress toward that object. Again visual feedback will be used, and this module will perform the image recognition necessary to accomplish those tasks. A more advanced implementation will use mapping algorithms instead of or in addition to the visual feedback.

Mapping Module - Becca

The Mapping Module will create a path for the hexapod to follow to reach it's target location. It will receive points from the Main Labkit FSM determining desired location, current robot location, and barrier locations . A series of heuristic algorithms will then create directions that will be sent to the hexapod via xbee. Feedback from the image recognition module will be used to determine how the hexapod is oriented. The orientation state of the robot will be stored by the labkit, updated continuously based on which direction the hexapod moves relative to which instructions were sent. If object tracking is successfully implemented, feedback will also be used to predict where the object is moving. Thus the hexapod will be able to track, instead of merely being reactionary.

Conclusion

Our project will use two FPGAs and be composed of two main parts. We will use a Nexys3 to control the movement of a hexapod, a six-legged walking robot. A camera mounted to the ceiling will be interfaced to a VGA monitor via the labkit FPGA, and the labkit will also determine, in various implementations, the direction and type of movement of the hexapod. The work will be split evenly by FPGA - Becca will work with the hexapod and the Nexys3 to control the actual movement of the robot, and Kelly will work with the labkit and camera to display video and use the labkit to control the hexapod. Both will work to interface the two parts together, and in the later stages both will work to improve the degree of complexity in the movement of the hexapod. If successful, our completed project will yield a fully functional hexapod which can move in various directions based on user input (push buttons), go to a specific point, and track another object.