Maseeh College of Engineering and Computer Science

PORTLAND STATE UNIVERSITY

Winter 2023

ECE544/558 - Final Project Proposal

Stephen Weeks, Drew Seidel, Noah Page, Emily Devlin
Email addresses: dseidel@pdx.edu, stweeks@pdx.edu, nopage@pdx.edu, emdevlin@pdx.edu

Project Description

For our Final Project, we are proposing to create a 2-wheeled differential drive robot and an accompanying Android App to control it. Our intention is to use our experience of implementing PID Motor Control using the Nexys A7 FPGA in a practical application and combine that with a sleek user interface conveniently located on a mobile phone.

The App will

- Provide the user with 4 buttons to control the robot's movement (Forward, Back, Right, Left)
- Communicate with the Raspberry Pi 4 over WiFi using MQTT to send and receive messages such as movement commands and the robot's status
- Provide the user with the ability to select the MQTT server and login credentials
- Stretch goals:
    - Visually display (i.e. with a gauge or a graph) information about each motor (speed, position, etc)
    - Receive and display a video feed from a Raspberry Pi camera mounted on the robot
    - Receive and display information gathered from the ultrasonic distance sensor

The FPGA will

- Use 2 Pmod HB3 modules to control the speed and direction of the 2 motors and use the motor's encoder information to implement PID Motor Control for consistent movement
- Communicate with the RPI using UART
- Stretch goals:

- Send motor status information (speed, position, etc) to the Raspberry Pi, which can send it on to the app for a user display
- Allow user interaction through push buttons and/or switches
  - For example, flipping a switch turns on or off the Raspberry Pi Camera
- Control or interact with an ultrasonic distance sensor
- Interact with IMU sensor(s)

Hardware Used

- Nexys A7-100T
- Raspberry Pi 4B 4GB
- 2x Pmod HB3
- 2x DC Geared Motor w/Encoder
- 4x 18650 Batteries 3000mAh
- 2x 18560 Battery Holder
- 2x Motor Bracket
- 2x Multi-Hub Wheel
- Ball Caster
- Acrylic Chassis
- HC-SR04 Ultrasonic Distance Sensor (stretch)
- Raspberry Pi Camera (stretch)

Tools Used

- Inkscape software to design chassis to be laser cut at EPL
- Android Studio for app development
- Vivado/Vitis for FPGA design and firmware development
- GitHub for version control and collaboration
- DMM, Oscope, and power supply for home testing/calibration

Design Approach

How do you plan to split the work?

Noah: design and build robot, HB3 motor control, integrating and testing

Drew: UART protocol development, develop/test/debug Android application

Stephen: design and program FPGA functionality sw/hw, help with UART protocol, integration and testing, FPGA stretch goals if time allows

Emily: Develop/test/debug Android application, work on Raspberry Pi camera stretch goal

Everyone: Debugging and testing, project demo, project report

How will you demonstrate success?

A successful project will achieve all of our committed goals. We will document our progress, challenges, and accomplishments in the project report.

What are your options if you start running out of time?

Possible ways to scale the project back:

- Demonstrate control of the components individually (motor/wheel assembly, camera) instead of integrating them together into a robot car
- Scale back the app to a single activity (as with ECE 558 Project 2)
- Simplify the options for user input and design more control of the wheels into the hardware (as with ECE 544 Project 2)

Demo Deliverables

Committed:

- FPGA:
    - Will setup motor speed and directions as hardwired constants for the system at startup
    - Will change direction via UART communication with the RasPi based on custom made packet transfer protocol
- Raspberry Pi:
    - Subscribes to MQTT messages sent by Android app; parses these messages and sends them on to the FPGA
    - Publishes MQTT basic motor information messages for the Android App to receive
    - Two-way UART communication with the FPGA:
        - Sends commands received from the Android App
        - Receives basic motor information or status updates
        - We will develop and map our own encoding scheme for these UART messages
- Android App:
    - Fragment Navigation with a Login/Settings screen and one main UI screen
    - 4 buttons (Left, Right, Forward, Back) the user can press to navigate the robot car
    - MQTT publish (button commands) and subscribe (motor info) communication

Stretch:

- FPGA:
    - Build a custom hardware module for the sonar sensor to send and receive pulses
        - Develop custom drivers to send and read data from the sonar sensor and convert to inches and feet
    - Allow for configurable setup with pushbuttons and switches to set motor speed
    - Display motor information to the 7-segment display for easy user visualization
    - Add an IMU sensor to the FPGA system to return motion data to the RasPi
    - Will send updated information of the motors to the RasPi

- Raspberry Pi:
    - Add Raspberry Pi camera to the robot car and stream the captured video stream to the Android App
        - We need to look into whether MQTT can handle a video stream or find an alternate communication method
    - Publish MQTT motor status messages for the Android App to receive
    - Add an ultrasonic distance sensor to the robot car and send information from this sensor to the Android App. Control of this sensor might be done entirely by the Raspberry Pi or might go through the FPGA (i.e. user presses a push button or flips a switch to turn on the sensor).
- Android App:
    - Info display screen showing information such as motor speed, position, etc.
        - Graphical display (instead of TextViews)
    - Video display screen streaming live video captured from a Raspberry Pi camera mounted on the robot car
    - Add ultrasonic distance sensor information to the Info Display Screen

A tentative high level block diagram of our committed project is the following:

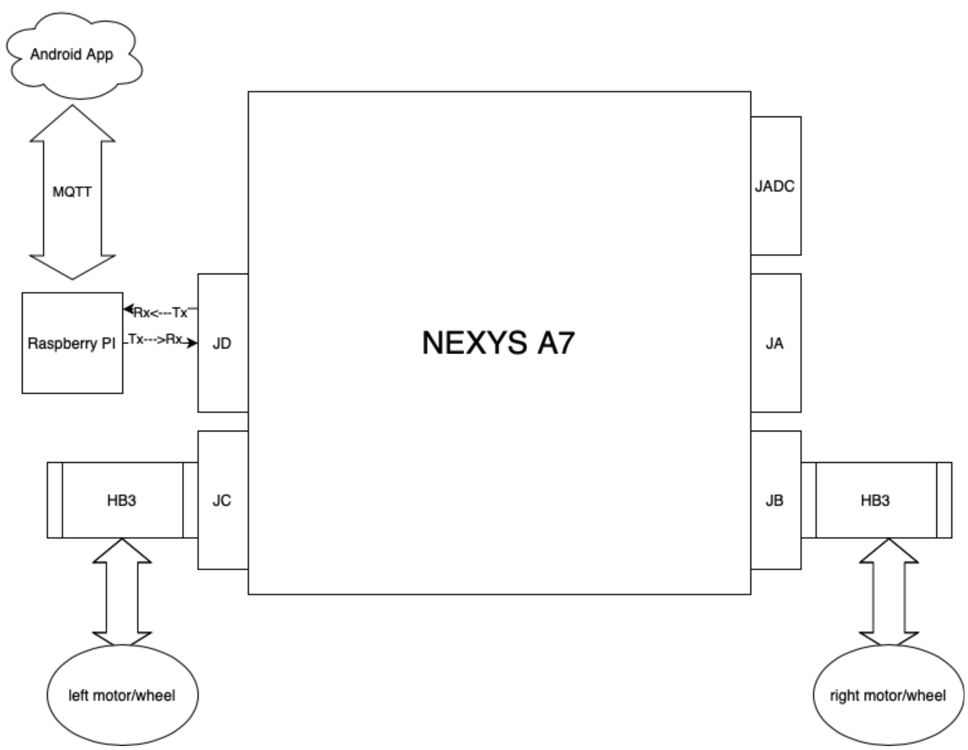

Figure 1: High Level Committed System Block Diagram

Milestones

3/10: Complete robot assembly
3/10: Complete initial structure of application (with placeholders)
3/12: Complete MQTT two-way communication in app
3/12: Implement PID control of each motor including both directions
3/14: UART communication between FPGA and RPI
3/16: Complete UI for app (display, etc)
3/16: Team meeting w/ Roy and Mehul
3/19: Additional stretch goals achieved/code freeze
3/20: Final Project Demo
3/23: Final Project Deliverables

Datasheets/Misc

Stretch Goals

Sonic sensor: https://cdn.sparkfun.com/datasheets/Sensors/Proximity/HCSR04.pdf, https://github.com/sparkfun/HC-SR04_UltrasonicSensor
Note: the sonic sensor will require a level shifter for output pins (5V supply)