EEL4914 ECE Design II
Final Report

SITH HAPPENS

by

Steven Paek
Daniel Suen

UNIVERSITY of FLORIDA

Department of Electrical and Computer Engineering
University of Florida

April 2019

Contents

List of Figures

Abstract

Our project is to build the "Star Wars R2-D2" using an RC Robot and Controller. The objective is to build a fun robot that people can play with that will behave like the robot from the Star Wars series. The roles for the robot and controller are described below.

For the controller, we have a physical controller board with multiple peripherals, and it will communicate with the robot via XBee. This controller will have a joystick to control the robot's movement. Additionally, it will have several buttons/controls to operate different functions of the robot such as audio output and other potential modes of operation. The controller will be equipped with an 16x2 character LCD that depicts the robot's status for the user. The controller will also run off of a set of AA batteries.

For the robot, a microcontroller will drive PWM signals to control the speed of the DC motors; by doing so, the robot's basic movement will be achieved. The control will be received via XBee from the controller. For aesthetics, the R2-D2 robot will have an LED circuit which will be controlled by the microcontroller to display different colors on RGB LEDs. For example, the LEDs will default to "Blue", just like the original R2-D2, but will adjust based on changing modes. When in use, the robot will play R2-D2 sounds from the movies. This robot will be battery powered, running off of a set of AA batteries.

Chapter 1

Project Introduction

1.1 Objectives

In order to meet our goal of developing a remote-controlled car, we require both digital and analog circuitry. To allow the user to control the robot, we desire various inputs to the system; these inputs come in the form of sensors such as buttons, switches, and an analog joystick. At the same time, the robot needs to be able to receive commands from the controller and exercise those commands in real-time. As such, an XBee module is attached to both the robot and the controller for wireless communication, and the microcontrollers can interact with the XBee modules using UART.

On the robot end, we will have PWM for motor control, an audio circuit including an external DAC to output sound effects from the Star Wars universe, and LEDs. Both the controller and the robot require batteries for power.

Our specific desired functionalities can be broken down as follows:

Digital Objectives:

- XBee UART Communication
- PWM Control for Motors

- SPI for the DAC
- SPI for the RGB LEDs
- Digital Control for the LCD
- Microprocessor Controls for R2-D2 and Controller (GPIO)

Analog Objectives:

- Battery Recharge Circuit
- H-Bridge Circuit
- Joystick Control
- Audio Amplifier Circuit

1.2 Technology Selection

The overall goal of the system is to have an interactive robot that can be entertaining. Initially, we were bouncing between two different options - using a traditional handheld remote controller or using a controller in the form of a glove with flex sensors.

We chose to implement a traditional handheld remote controller to operate the R2-D2 robot because it would be more intuitive to use, and the form factor was a big factor behind our decisions. This choice caused us to abandon the use of flex sensors and accelerometers and instead focus on more classical sensors such as digital buttons, switches, and a joystick.

Additionally, we chose to drive the robot using 4 DC motors connected to wheels. We decided to implement proportional steering and throttle because again, this would be the most intuitive to the end user. Ultimately, we also decided to put an LCD on the controller in order to update the user with the status; this also ended up helping when we were working to debug our code.

Chapter 2

Analysis of Competitive Products

There are existing R2-D2 toy robots with different capabilities and price ranges. For example, Disney sells an R2-D2 figurine that spins, lights up, and makes realistic Star Wars sounds. Additionally, Hasbro also markets a similar robot that is a controlled by a user phone application and can dance to prerecorded music.

Figure 2.1 R2-D2 Figurine, sold by Disney

Figure 2.2 R2-D2 Robot, sold by Hasbro

These products provide functionality that is entertaining to the user. However, the first option lacks a fully manual driving mode while the second option is expensive. Because we are also limited on time and would like to focus more on a hardware design, we choose to implement a hardware controller rather than develop a software app for a mobile phone. This also allows us to have more flexibility in that we can control the robot wirelessly without using other protocols such as Bluetooth or WiFi.

Chapter 3

Project Details

3.1 Project Architecture

The block diagram is divided into two parts: the controller and the R2-D2 robot.

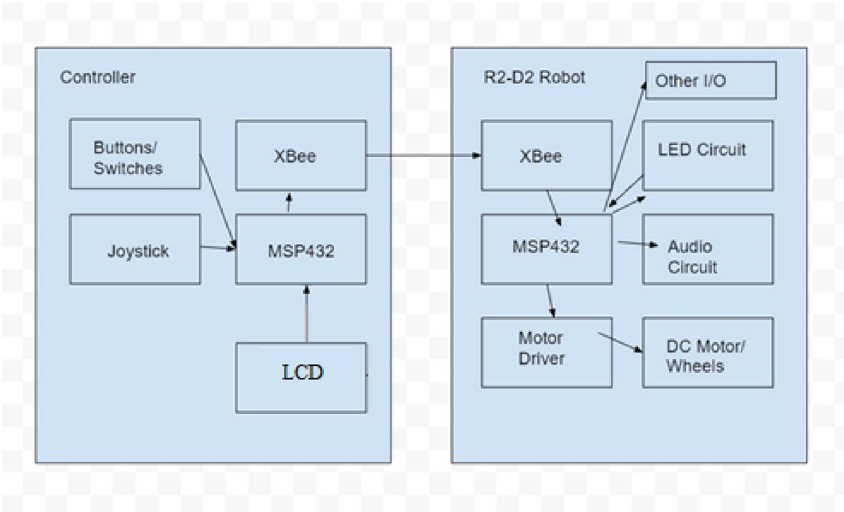

Figure 3.1 Block Diagram

The controller is powered by a microcontroller (MSP432), and it accepts different inputs including digital buttons, switches, and an analog joystick. The digital buttons and switches are for controlling which mode the robot is in (party/manual mode), motion, and sound. The analog joystick is used for determining which direction to move the robot. These control signals are sent via XBee wireless communication to the R2-D2 robot. Additionally, the MSP432 on the controller end updates an LCD screen to inform the user of the robot's status.

The R2-D2 robot is also powered by the same microcontroller (MSP432). On the robot end, the XBee is used to receive the different user inputs, and each byte is parsed in code to determine the received data. Then, the MSP432 is used to drive various functions on the robot itself by communicating data via SPI to the LEDs and the DAC for audio, and sending PWM signals to the H-bridge circuit for movement (which are sent to the robot's wheels).

3.2 Hardware Selection

3.2.1 Microcontroller

Because the project consists of both a controller and a robot, we need two separate microcontrollers with the ability to communicate with each other. Since we do not require a microprocessor with specific capabilities, we select the MSP432 for both the ease of use and its versatility. The MSP432 served as the microcontroller on both the controller and robot sides.

This specific microcontroller was selected because it is low-power, relatively inexpensive, and provides much of the desired functionality (including UART, an ADC with sufficient resolution of 14 bits, built in SPI, and multiple timer channels for PWM) for our purposes. There is also sufficient GPIO for the task at hand. It can run at 48MHz as well, which is good when we are running computationally intensive tasks at the same time.

- MSP432: Up to 48MHz CPU, FPU, 14-bit ADC, UART, 16-bit Timers with PWM (Robot/Controller)

3.2.2 Wireless Communication

The XBee module series 2 is used to have two microcontrollers communicate with each others wirelessly through UART. This type of communication was selected because it is easy to use and interface with. We connect each module to 3.3 V and ground, and then connect the “data out” to the RX pin and the “data in” to the TX pin for the same UART channel on the MSP432. It works the same way as a wire-to-wire transmission would. We also configure the XBee to be talking on the same channel with destination and source addresses set appropriately.

- XBee: RF Module IEEE 802.15.4 Standard, UART Communication (Robot/Controller)

3.2.3 Audio Circuit

To output sound from the microcontroller on the R2-D2 robot side, we use the MSP432 to send signals to a 10-bit external DAC (LTC1661). The output of the DAC is sent through a Class AB Amplifier circuit before sending it to a speaker. The LTC1661 was selected because we are both familiar with the chip, and it supports sufficient resolution (10-bit) for audio.

- LTC1661: 10-bit External DAC, 2.7V to 5.5V Supply (Robot)
- Audio Power Amplifier Circuit (Robot)

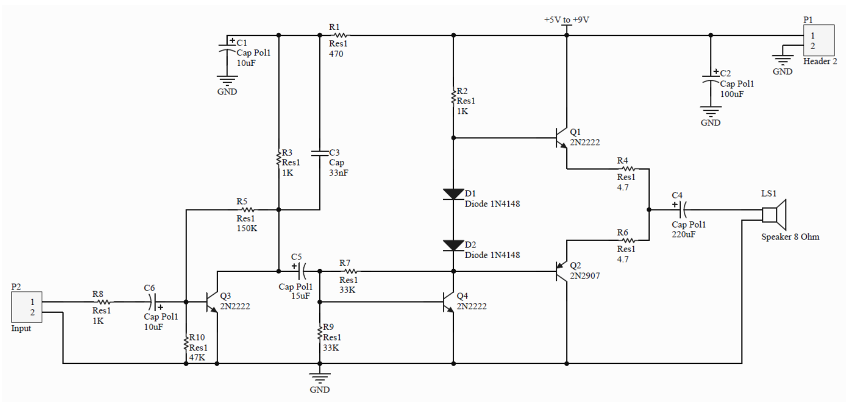

Figure 3.2 Class AB Amplifier

3.2.4 Motor Driver

The motor driver circuit consists of an H bridge to drive four motors for movement. This is essentially a switching circuit, which controls how often voltage is allowed to be supplied to the motors. By changing the PWM duty cycle, we can change the total amount of power, thereby changing the speed. When the left and right motors turn at different speeds or even different directions, we will achieve turning or pivoting motions.

- H-Bridge Circuit (Robot)

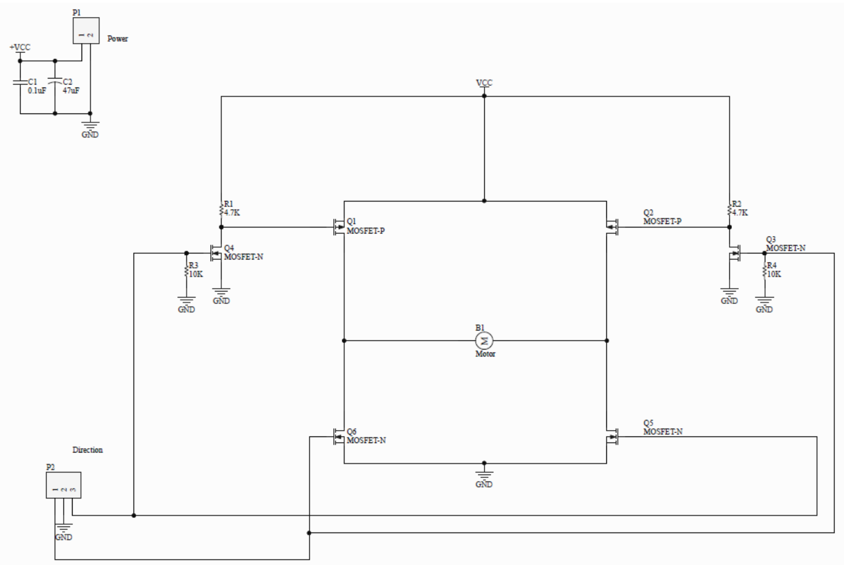

Figure 3.3 H-Bridge Circuit

3.2.5 LED Circuit

We included the use of the APA102 LEDs, and there are 32 bits used to control an LED. We send this data via SPI, and from there we can adjust the R, G, and B registers to create an overall color. These were selected mainly for the ease of use.

3.2.6 Controller Inputs

The following digital buttons and switches and analog joystick will serve to provide user input to the system for robot control.

- Push Buttons
- Switches
- Joystick: Analog Joystick (Controller)

3.3 Software

We include rough flowcharts to describe the functionality of the software design at a high level. The first flowchart describes the functionality of the controller.

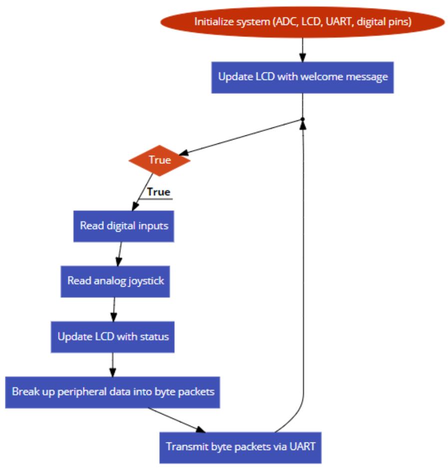

Figure 3.4 Controller Flowchart

At the start, the controller initializes all of the systems including the digital ports, UART, LCD, and ADC. The software consists primarily of a while-true loop in which the program alternates between reading the analog joystick pins and the digital inputs and then making decisions. The analog joystick determines which direction to move the robot while the digital inputs are used to determine motion, sound, and mode. These commands are all sent via XBee to the R2-D2 robot. The transmission happens inside the while loop.

The second flowchart describes the functionality of the R2-D2 robot.

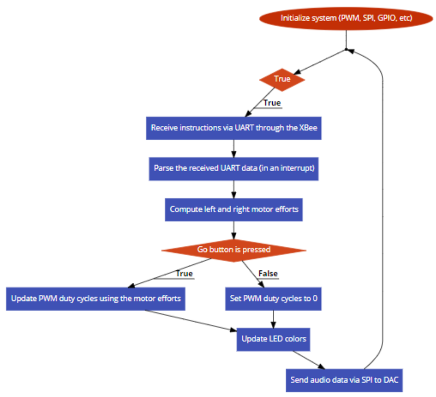

Figure 3.5 Robot (R2-D2) Flowchart

The software for the robot initializes all the necessary systems and consists of a while-true loop and several interrupt service routines. In the loop, the program receives the instructions from the controller via XBee and then reconstructs the data. The UART data is received in an interrupt service routine, and the byte data is parsed in the ISR. We compute the left and right motor efforts from the joystick ADC values. If they exceed a certain threshold and a button is pressed, the PWM duty cycles are changed, allowing the robot to move. Meanwhile, the robot also plays sounds and changes LEDs in the background.

Chapter 4

Design Procedure

Chapter 5

User Manual

Chapter 6

Bill of Materials

Major Part	Vendor	Quantity	Unit Price	Part Price

Chapter 7

Work Responsibilities and Gantt Chart

7.1 Work Responsibilities

For the project, Steven Paek was assigned to be in charge of the R2-D2 robot, and Daniel Suen was assigned to be in charge of the controller. As such, the work responsibilities were divided as follows:

Steven Paek	**Daniel Suen**
R2-D2 PCB	Controller PCB
Motor PWM	Battery Recharge PCB
Audio Circuit with External DAC	Joystick and Button Control
LED Circuit	Controller LCD
Other I/O	XBee Communication

7.2 Gantt Chart

The Gantt chart outlines the proposed timeline for completing the project and meeting our own internal deadlines. There was a focus on completing the most of the PCB designs and analog circuitry towards the beginning of the semester.

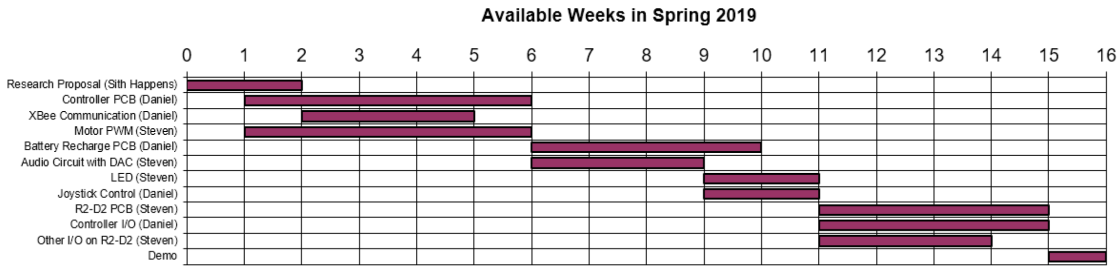

Figure 7.1 Gantt Chart

Appendix