Final Project: RC Car

Ryan Baker
Rowan University

December 21, 2018

1 Design Overview

In this milestone project, the MSP430G2553 and the MSP430F5529 were used to make an RC car and controller. The car had to be controllable over Bluetooth using the controller. A potentiometer was used on the controller to act like a steering wheel. The servos on the car had to respond to the position of the potentiometer and turn at various levels accordingly. This had to be done by sending an ADC value over Bluetooth using UART and then changing the PWM duty cycle of the servos based on the value received over UART. By the end of the project, a fully functioning RC car and controller were created such that the car were turn based on the position of the potentiometer on the controller.

1.1 Design Features

The design features of the controller:

- Read a voltage in from a potentiometer
- Process the data
- Send the processed data to the Bluetooth module using UART

The design features of the car:

- Receive data from the Bluetooth module using UART
- Calculate the new speed of each servo based on that data
- Adjust the PWM signal to each servo accordingly

1.2 Featured Applications

These are the featured applications:

- Entertainment, or use as a toy
- Use for hobbyists
- Access to small spaces

1.3 Design Resources

$https://github.com/RU09342-F18/intro-to-embedded-final-project-me$

1.4 Block Diagram

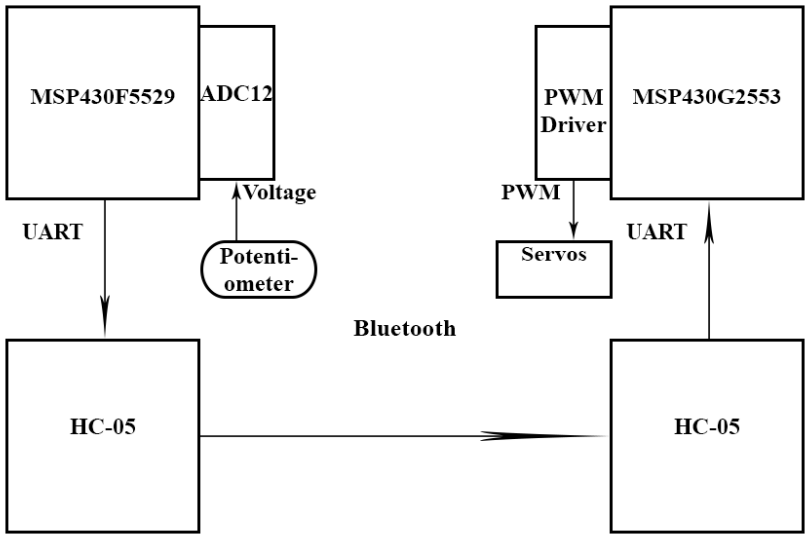

Figure 1: Block Diagram for RC Car

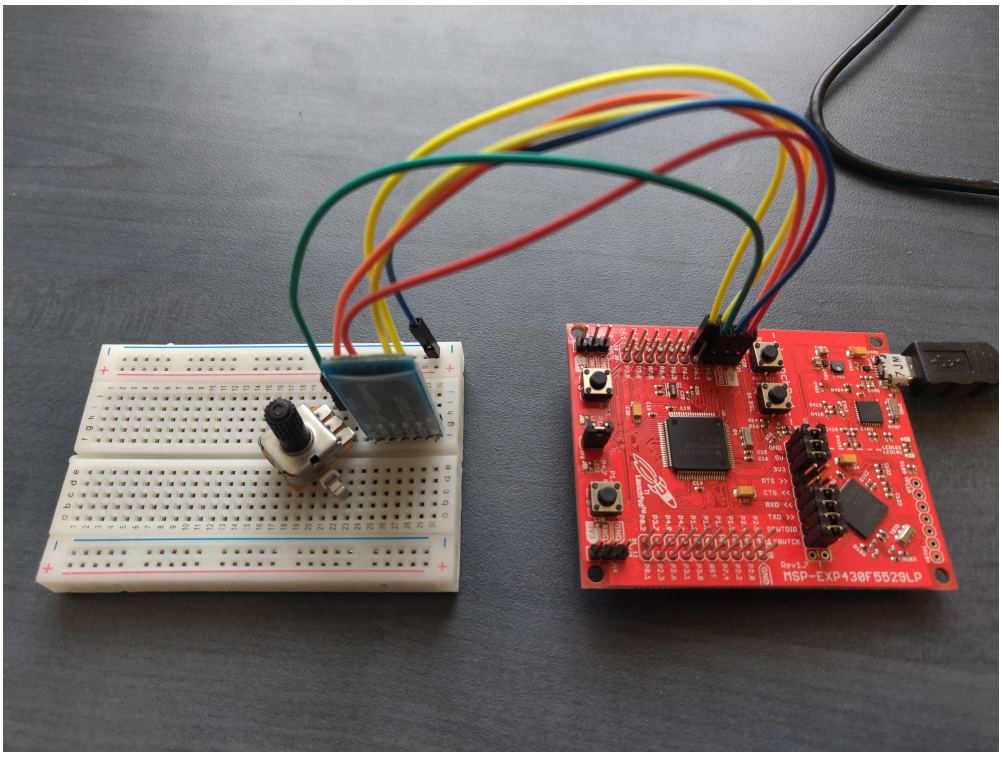

Figure 2: Board Image

1.5 Board Image

2 Key System Specifications

PARAMETER	SPECIFICATIONS	DETAILS
Input Volt.	0 - 3.3V	The voltage into the ADC due to the potentiometer
PWM Duty Cycle	5 - 10%	Duty cycle of the two pairs of servos

3 System Description

This system is designed to be an RC car with a separate wireless controller. The task is to use both an MSP430G2553 and an MSP430F5529 to create a car and controller that are connected via Bluetooth. The user can control the car using the controller.

3.1 Highlighted Devices

- MSP430G2553 - This device was used to receive over UART and generated PWM signals
- MSP430F5529 - This device was used to read a voltage using the ADC and transmit over UART

3.2 MSP430G2553

The MSP430G2553 is used in this system as a PWM driver and a UART receiver. The PWM driver is used to control the four servos in the car. The PWM signal has to have a duty cycle of 5 - 10% and have a period of 20ms. The UART receiver is used to get the inputs from the controller from the Bluetooth module. The inputs from the controller are scaled appropriately and used to modify the duty cycle of the PWM signal.

3.3 MSP430F5529

The MSP430F5529 is used in the system as an ADC (analog to digital converter) and a UART transmitter. The ADC is used to convert the voltage across a portion of the potentiometer into digital values that can be sent over UART. The input voltage has a range of 0 to 3.3V. The UART transmitter is used to send the value from the ADC to the car. The value from the ADC is shifted to fit in a byte before being sent over UART.

4 SYSTEM DESIGN THEORY

For this system, the desired inputs are sent to the controller as a voltage using a potentiometer. That voltage is stored as a 12 bit value. The 12 bit value is shifted right six times since UART sends a byte at a time and the voltage no longer oscillates when the potentiometer is stopped if the resolution is scaled down to 6 bits. The byte is sent to the Bluetooth module using UART and it is transmitted wirelessly to the other Bluetooth module. On the other board, the UART signal read in is determined to contain a value of less than or greater than 32. This correlates to the potentiometer being turned left of center or right of center. Depending on if the car needs to turn left or right, the value will directly scale the PWM signals of the left and right servos so the position of the potentiometer affects how drastically the car turns.

4.1 Steering Sensitivity

The goal of this project was to have the steering of the car have a higher number of unique turning speeds. Theoretically, with a byte being sent over UART, there should be 127 unique levels of turning acheivable for each direction. However, do to the value from the ADC oscillating even when the potentiometer is no longer moving, there are

realistically only 31 unique levels of turning for each direction. In the end, this is still higher sensitivity than I've seen with other solutions in this price range. Additionally, a future improvement could be to use capacitors and inductors to smooth out the voltage from the potentiometer so more unique values are achievable.

5 Getting Started/How to use the device

To use these devices, power on the controller using the included USB cable by plugging it into a computer, outlet, or battery. Power on the car by plugging the board into the portable phone charger and the servos into the generic RC car battery. Then give the controller and car about two to three seconds to connect and you can begin controlling the car with the controller.

6 Getting Started Software/Firmware

This section will discuss program each of the boards in the project.

6.1 Device Specific Information

MSP430G2553: Create a new project in Code Composer that uses the MSP430G2553. I used CCS 8.2.0 for this. Add the code to your project by adding the file directly or copying and pasting the contents of the file into a new file. Debug and let CCS flash the code to your processor. If there are any updates, update your board. Once the code is running on your board, you can unplug it and move on to the hardware portion of the setup.
MSP430F5529: Using Energia, open the sketch from the repo, or copy and paste the contents of the sketch into a new one. I used Energia 1.8.7E21 for this. Go to 'Tools' $>$ 'Board:' and select the MSP430F5529. Then, go to 'Tools' $>$ 'Port' and ensure that the COM port of your MSP430F5529 is selected. Next, click the upload button near the top left of the window. After the code has been flashed to the board, it can be unplugged, and you can move on to the hardware portion of the setup.

7 Test Setup

To test these devices, power on both the controller and the car as described in Section 5. Once the controller and the car have been paired, rotate the potentiometer to various positions and check to see if the vehicle has turned at the expected rate and in the expected direction. For further testing, ensure that the controller is connected to your computer for its source of power. Follow the previous steps, but now open the serial monitor in Energia to see if the values been sent over UART are expected.

8 Design Files

8.1 Bill of Materials

- MSP430G2553
- MSP430F5529
- 10k ohm potentiometer
- 2x HC-05 Bluetooth module
- 4x FS90R continous rotation servo