# ECE 511 Microprocessors Group Project

*Proposal for a Copy Cat Remote Control Car System*

***Group 3***

***September 23th 2013***

## *Purpose*

The purpose of this project is to modify a commercially available Remote Control (RC) car so that the user can operate the vehicle in three distinct modes: Run, Record and Playback. The operator will send commands to the car using a custom remote control. In Run mode commands will be passed straight through to the RC car. In Record mode the operator's inputs for a specified period of time will be recorded. In Playback mode the pre-recorded operator inputs are played back with the option of replay.

## *Background*

RC cars are typically operated by passing signals from a User Input Device (UID) (often a trigger for acceleration/ breaking/reverse and a wheel for steering) through a radio transmitter to a receiver located at the car. This receiver then disseminates the Pulse Width Modulated (PWM) control signals to the motor/engine controller and the steering servo.

Hobby level RC cars are driven be either a powerful electric motor or a small combustion engine. In the case of a combustion engine, a servo controls the throttle and break. In the case of the electric motor an Electronic Speed Controller (ESC) drives the electric motor by providing the appropriate current to the electric motor. Electric cars rarely have dedicated breaks. Most RC car steering use a simple servo connected by mechanical linkages to the front wheels.

## *Theory of Operation*

The MSP430 will act as interface between a simple wired UID and the vehicle. The MSP430 will interpret the signals coming from the UID. In Run mode the MSP430 will translate the input signals into PMW control signals and send them to the actuators. In record mode the MSP430 will store them for later use, translate them for output and send them to the actuators. In playback mode the MSP430 will ignore actuator inputs and translate the stored signals and send them to the actuators. The user will be able to toggle through these modes at will.

The MSP430 has limited flash and RAM, these will be needed to store and facilitate the execution of our code. In order to store the input signals during the Record mode, our system will require an encoding scheme to compress the input data. This encoding will record when and how the input changes as opposed to simply storing the input signals. This compressed data will need to be translated into the PWM control signals that the actuators require. Input Buffers and output buffers will be designed so

inputs and outputs won't be missed due to encoding or translation delay. The MSP430's timers will be used to generate the PWM signals.

If time permits we will attempt to build a more sophisticated UID. Rather than relying on a wire connection, this UID will communicate using Bluetooth. This will require connecting a Bluetooth module to the MSP430 and programming the necessary code to facilitate serial communication.

Since the MSP430, and any external elements attached to it, will be located on the RC car they will needed to be mounted securely and potentially shielded against operating vibration. In addition the operator will need a visual indicator of what mode the RC car is currently in, this will also be located on the car. The MSP430 and any additional components will need either a dedicated power supply or will need to share the RC cars onboard battery.

## *Components*

The heart of our system is the MSP430 G2553IN20 mounted to the LaunchPad development board. This 16-bit microcontroller contains 16kB of Flash and 512B of RAM. It also has 16 general purpose IO and a Universal Serial Communication Interface (USCI) module. The system is low voltage, operating from 1.8 V to 3.6 V. The onboard timers will be used to facilitate the transmission of PWM signals.

The HPI Racing 105502 Mini Recon 2.4 RTR will serve as our RC vehicle. This $1/18^{th}$ scale truck is driven by a brushed DC motor. The RC truck is powered by a 7.2 V rechargeable battery. It includes an ESC separate from the receiver. This is important because since we are not using the receiver it can be isolated from the on board power supply, which will improve battery life. The MSP430 is connected directly into the steering servo and ECS control inputs respectively.

The UID will come in two varieties, basic and advanced. The Basic UID will consist of five push button switches. These switches will each control a different input. The basic inputs will be forwards, reverse, left, right, and change state. The button circuit will be connected to the MSP430 by a ribbon cable. The Advanced UID will be a simple smartphone application that will communicate with the MSP430 via a Bluetooth connection. A DFRobot Serial Bluetooth module will need to be connected to the MSP430 in order to facilitate wireless connectivity. We will use the USCI to communicate between the MSP430 and the Bluetooth module.

The remaining major component is the power supply and mode indicators. The power supply will need to regulate the 7.2 V from the on board battery down to the 3.3 V the MSP430 and either the wired controller or the Bluetooth module can be driven by. This circuit should also have some form of over draw protection limiting the current to safe levels. In addition the mode indicators will be a simple set of three different colored LEDs.

## *Stages of Development*

The first stage of this project is simply designing the input interface to the MSP430 and the output interface to the RC car. This will require analyzing the ESC and steering servo to determine the range of values that drives each of them. Also we will explore the feasibility of interfacing the MSP430 to the

Bluetooth board. In parallel the interface to the basic UID will be designed. Equipped with the PWM values obtained in the RC car analysis, the PWM output will be designed in the MSP430. The completion of this stage will allow a user to pass input values via button presses to the MSP430 and to pass predetermined PWM signals to the RC car. An objective in this stage is to connect a device to the MSP430 via Bluetooth.

The second stage will require designing the encoding scheme to store the input button press durations. Once the encoding method is developed it will need to be programed into the MSP430. Also after the encoding scheme is completed a translation to PWM duration must be developed. This translation must be programed into the MSP430. The completion of this stage will allow a user to encode a series of button press inputs and separately play them back via translation to the RC car. This represents a limited Record mode without run through and a rough Playback mode. An objective of this stage is to pass button press data to the MSP430 via Bluetooth.

The third stage will require designing the input and output buffers that will facilitate real time operation. The input buffers will be integrated between the input interface and the encoding routine. The output buffer will be integrated between the translation routine and the PWM output. A location to store the current mode will also be identified. Once this stage is completed the first real time operation of Run mode and Record will be possible. An objective for this stage is that a simple five button controller application can be implemented on a smart phone and button signals can be sent to the MSP430 via Bluetooth.

The fourth stage of this project will integrate the three operating modes together and real time operation will be refined for performance. Interrupts will be designed to allow the user to switch between the different operating modes. Separate interrupts will be used for each mode and an interrupt to ‘replay/restart’ in Playback mode. At the end of this stage the user will be able to toggle through each mode and use all mode functionalities without noticeable latency or signal error.

The fifth and final stage will address the power requirements, indicators and packaging. The power supply of RC car will need to be modified to support MSP430, the UID (either wired controller or Bluetooth module) and the mode indicators. Finally, all the components (RC car, MSP430, LEDs, etc) need to be firmly attached and contained so that a user can drive the RC car and cycle through the three modes without inconvenience.