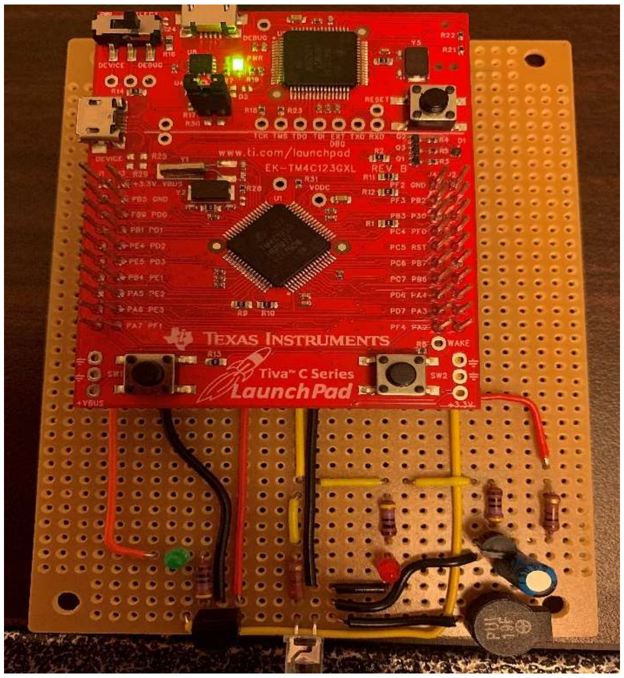
Introduction:

As a project for Embedded 1, I built a remote controller prototype using the TM4C123GXL microcontroller (RedBoard). The device serves as a bidirectional IR interface capable of learning and playing NEC format commands. The components used for the project are: GP1UX511QS 38KHz IR detector, IR Led, Green Led, Red Led, 470-ohm resistors, 220-ohm resistors, AT-1127-ST-2-R transducer, 10 uF capacitor, 2N3904 speaker driver transistor, 10 mil pitch unshrouded header, wires, and a PC Board.



Theory of Operation:

The IR detector sensor is powered by 3.3V from the RedBoard. GPIO pin PA2 is set up as an input which is connected to the sensor output. GPIO pin PB5 is set up as output and is connected to the IR Led. When a remote button is pressed, the IR signal sent by the remote is captured by the sensor and sent to the RedBoard through PA2. The captured signal is then analyzed at different time periods using edge triggered interrupt and Timer interrupt on the RedBoard to recognize which button was pressed. Then, the data and address bits of the signal are recorded in EEPROM. To send an IR signal that matches the signal sent by a remote button press, an alternate function called “pulse width modulation” is set up to control the PB5 pin. The output signal sent by IR led is then modulated to match the signal sent when a button is pressed on the remote. PB5 is also connected to a speaker circuit for convenience to recognize a remote button press. If an IR signal using NEC format is received with any errors, bad alerts are enabled to output a lower pitched tone that indicates an error, whereas if IR signal received has no errors, a higher pitch tone can be enabled to indicate success.

The device uses UART interface to transmit commands and receive output. The following commands are supported:

1. decode - Waits to receive NEC remote signal press and then displays the address and data of the remote button pressed. (Optional) plays a “good”/ “bad” tone when NEC alert is enabled as shown in no.7

2. learn NAME - Receives an NEC IR command and stores as NAME with the associated address and data in EEPROM (e.g. learn ch-)

3. erase NAME - Erases command NAME by setting the valid/invalid to 00 0r 01 in EEPROM (e.g. erase 1)

4. info NAME - Displays the address and data of the command NAME (e.g. info eq)

5. list commands - Lists the stored commands

6. play NAME1, NAME2, … NAME5 - Plays up to 5 stored commands (e.g. play ch-, 1, eq)

7. alert good|bad on|off - Turns the control the alert tone on and off for good and bad received IR commands (e.g., alert good on, alert bad off)

Conclusion:

This project was very fun. Concepts regarding how serial communication such as UART works, how interrupt works, how to modulate a signal, and how GPIO pins can be calibrated to input and output different types of signals on a microcontroller needed to be used for this project, and I have got a good understanding of those due to this project. This was also my first time programming a microcontroller, and it has made me realize that these tiny devices are very cool and powerful.