BAssist
(Banjo Assist Robot)

Fatemeh Gholizadeh
David Hatch
Shiva Khanal
Gavin Philips

Abstract

Playing musical instruments is a specialized skill requiring years of practice and dedication to master. For some people, this commitment is not an easy one. For those with medical conditions affecting use of their hands, it may be impossible. The Banjo is one such instrument.

The Banjo is played with two hands, one hand to finger the frets of the desired string, the other to pluck the string sounding the note. For those not capable (or willing) to operate with both hands simultaneously, the Banjo Assist (BAssist) Robot will handle plucking the strings in preprogrammed patterns, allowing the user to play banjo only by fretting the desired notes.

The BAssist uses a TI MSP430F5529 and five solenoids to pluck each string, assisting in the playing on the banjo in the Scruggs style, which is defined by a rolling pattern of picking alternate strings, typically in a pattern of eight which is used as the length of the pre-programmed patterns. In addition, BAssist allows the user to control tempos via push buttons and review information about the patterns on an LCD character display peripheral. This system will hopefully give those who were previously incapable of playing any two-handed instrument the ability to play the banjo.

Table of contents

Contents

Motivation

Music is enjoyed by many people around the world, but the joy of creating music is experienced by a much smaller number. Playing a musical instrument can be difficult for many people, and for some, it may be impossible to play certain instruments the way they were intended.

One such instrument is the Banjo. With many styles of playing, each requiring the synchronous use of two hands, this instrument can be tricky to learn or almost impossible to play without full use of both hands. One style of playing is the Scruggs style. This style is characterized by a rolling pattern of alternating string plucks, most commonly patterns of eight. These patterns are played with the plucking hand while the other hand frets the desired chord.

Solution

BAssist aims to simplify the playing of the banjo in the Scruggs style by allowing the user to perform using only one hand to fret the desired chords. BAssist, which is implemented on a TI MSP430F5529 microcontroller, will pluck out pre-programmed patterns of 8. The patterns are set to some of the most common patterns beginners in the style learn, and BAssist will play the patterns at a tempo set by the user via the tempo control system.

The implementation of this system will open up the possibility for those to perform on the banjo without needing to worry about controlling two hands simultaneously, which may also allow those who may not have full control of two hands to experience the joy of playing a musical instrument.

Block Diagram

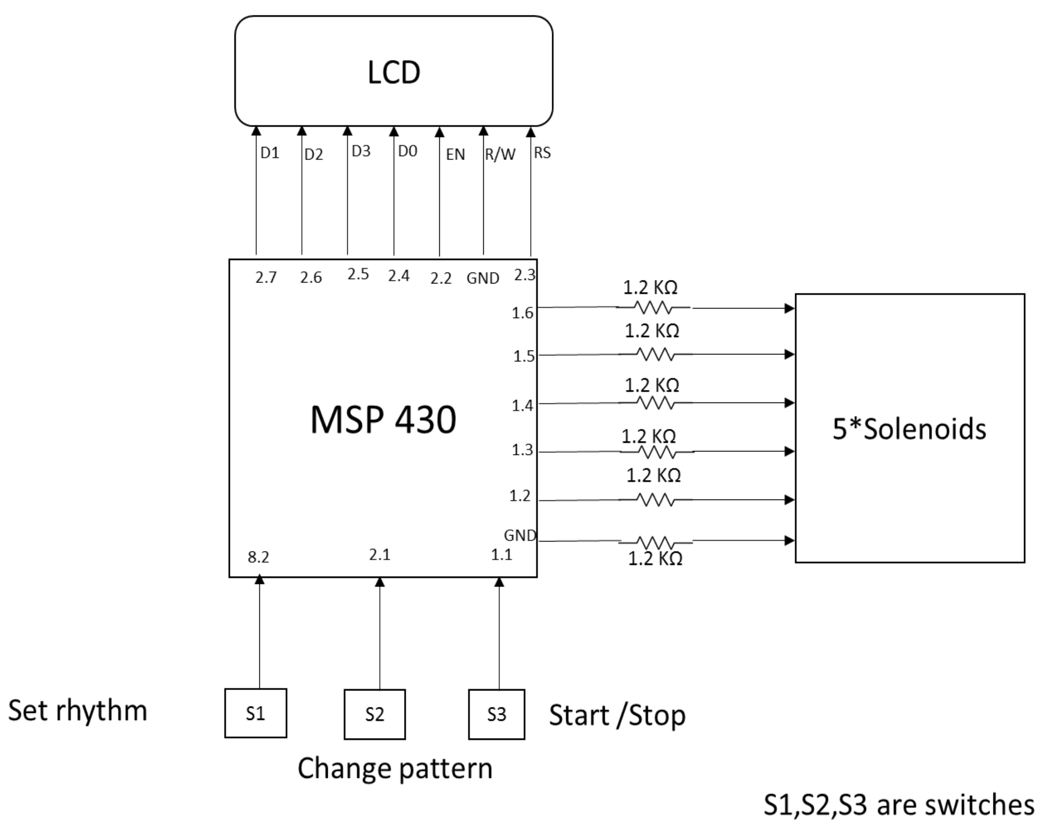


Fig. 1: Block Diagram

Components

Solenoids

Fig. 2: 5V Mini push pull Solenoid

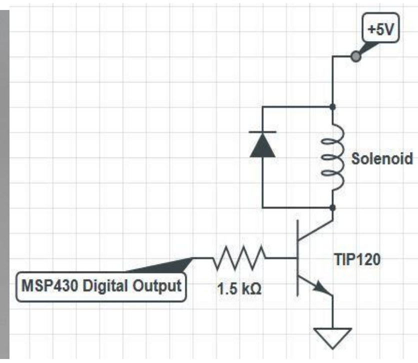


Fig. 3: Interface connection

Description:

Solenoids are electromagnets – they are made of a big coil of copper wire with an armature (a slug of metal) in the middle as shown in Figure 2. When the coil is energized, the slug is pulled into the center of the coil. This allows the solenoid to push and pull via the application and removal of power.

Hardware interface:

Solenoid actuators connected to GPIO pins through TIP120 Darlington power transistors as shown in Figure 2. Note that a 1.5 kOhm resistor is placed in series with the base of the transistor to limit current between the MCU’s output pin and the transistor.

Software interface:

The software interface for the solenoids is simple. Upon board initialization, five GPIOs (one for each banjo string) are configured as output pins. Because the solenoid can only move between two states, each solenoid can be completely controlled with one digital general-purpose I/O pin. When the application needs to move the solenoid, it toggles the line.

LCD Character Display

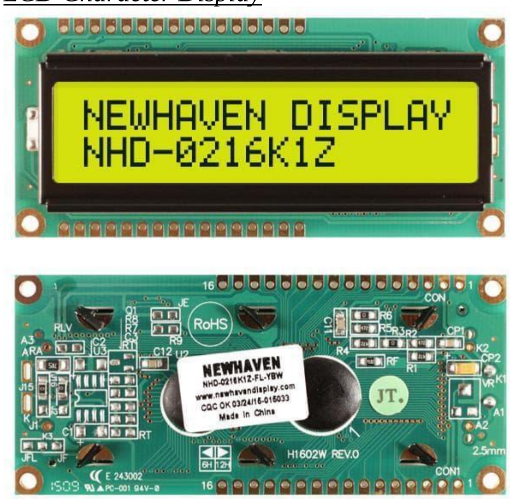


Fig. 4: Newhaven Display LCD Character Display NHD-0216K1Z-FL-YBW

Description:

As its name would suggest, the LCD character display is used for displaying numbers and characters. In BAssist, the application sends pattern names to the screen to inform the user what pattern he has selected.

Hardware interface:

Hardware interfacing for BAssist's LCD character display is relatively complicated. The LCD character display uses 5V logic, but the MSP430F5529 utilizes 3.3V logic. Unfortunately the MCU's logic high is below the threshold for a logic high on the LCD character display, which is $V_{IH} = (0.7)*V_{CC}$, where V_{CC} is 5V. This corresponds to $V_{IH} = 3.5V$. The two data sheets provided for the LCD character display apparently contradicted each other. One suggested that the display had to be operated at 5V, but the other suggested that V_{CC} could range from 2.7V to 5V. Only 5V was tested.

Consequently, there are logic level converters for each of the control and data lines shared between the MCU and the LCD character display. The logic level converters step up the logic highs from 3.3V to 5V. To reduce the number of MCU GPIO pins, the MCU and LCD character display share only 4 of 8 possible data lines.

Though it does not directly interface with the microcontroller, the LCD character display requires an additional supply voltage at ~1.3V for character contrast. A voltage divider consisting of a 3 kOhm and a trimmed 1 kOhm resistor between 5V supply and ground provide the appropriate voltage to the display.

Software interface:

While the hardware interfacing with the LCD character display was less than ideal, the software interfacing was even hackier. The datasheet for the NHD-0216K1Z-FL-YBW specifies a list of commands used for controlling the display. The MCU sends initialization commands to inform the LCD character display that only 4 data lines will be used. After the initialization commands, the application firmware can write up to 32 characters to the display.

The LCD character display does not use a standardized form of serial communications such as UART, SPI, or I2C. Instead, the firmware implements a custom driver that manipulates GPIO lines. The display manufacturer provided pseudo-code driver functions, which were implemented in the firmware. The various lines have to meet timing requirements for the commands and data to be correctly processed by the character display, so a 1 kHz timer was implemented in the firmware to ensure consistency in timing the line changes.

<u>Switches</u>

Description:

A switch allows the user to interface with the system manually. Switches on BAssist control tempos, pattern selection, and whether the system is playing. BAssist uses both onboard and offboard switches. Broadly, switches encode two possible states – opened and closed.

Hardware interface:

Each switch is connected to one GPIO on the MCU. The other end of each switch is tied to ground, and current limiting resistors are included to prevent shorting the MCU pin to ground.

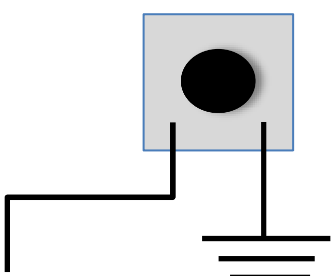


Fig. 5: switch

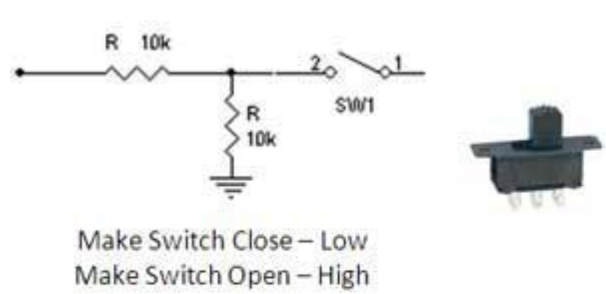


Fig. 6: internal connection.

Software interface:

Though the hardware interface for the switches is very simple, the software is a bit more complicated. Because the user controls the switch state, the application firmware must configure the GPIOs as inputs. Additionally, the application firmware does not know when the user will change the switch state. Polling the switch is inefficient, as users do not frequently manipulate the switches relative to the speed of the MCU. Therefore the GPIOs associated with the switches are associated with interrupts, which change the selected pattern, the tempo, and the play state.

There are also issues associated with the electrical connection when the user depresses the switch. Because the contacts may connect and disconnect several times in a very short interval, there may be multiple edges triggering multiple interrupts. To avoid unintended interrupts, the application firmware uses the 1 kHz timer to ignore edges for a short time after hitting the interrupt.

Results

Overall, BAssist is a success. The user can cycle through the different patterns, enable and disable play, and control the tempo at which each pattern is performed. The user sees the pattern name on the LCD as he cycles through the different patterns as well.

BAssist does have some limitations. Solenoids draw a tremendous amount of power, and much of that power dissipates as heat. If a user accidentally touches the solenoid after using BAssist for a long time, he may find himself exclaiming “yowch!” and searching for the nearest ice water. Additionally, fast tempos cannot be played reliably on the BAssist. Finally, the hardware encasement was not reliable and robust to banjo movement.

Bibliography

Solenoid datasheet and information on hardware interfacing:
https://www.adafruit.com/product/412
http://playground.arduino.cc/uploads/Learning/solenoid_driver.pdf

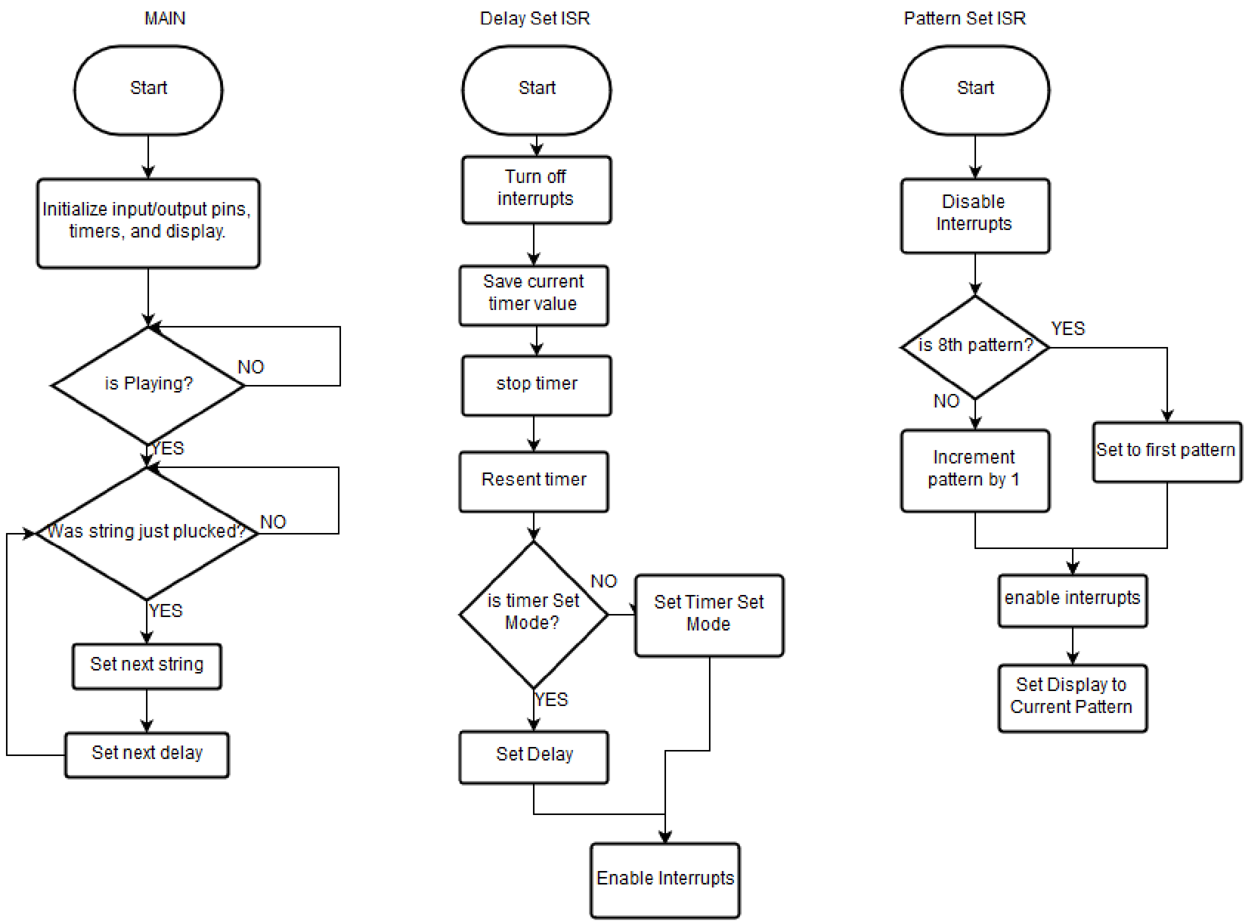
Newhaven Display Datasheets:
http://www.newhavendisplay.com/specs/NHD-0216K1Z-FL-YBW.pdf
http://www.newhavendisplay.com/app_notes/ST7066U.pdf

Appendix

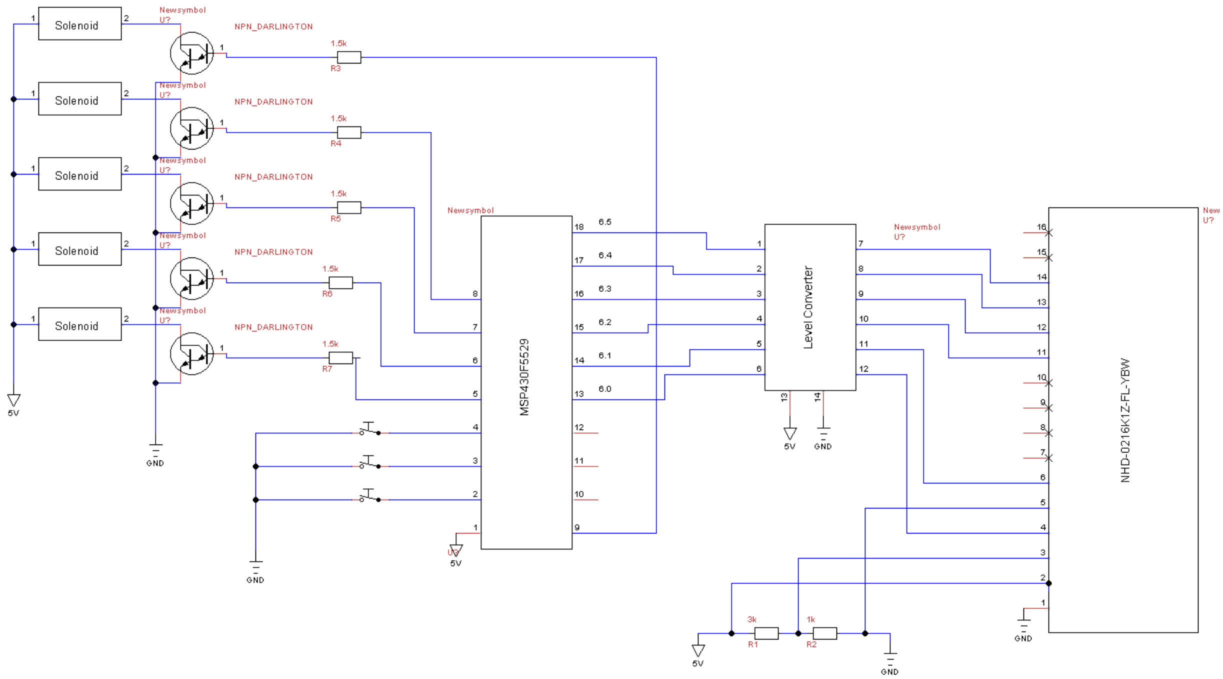
Team Members/Task Division

Fatemeh Gholizadeh: LCD hardware interfacing and firmware development
David Hatch: solenoid-banjo interfacing, component selection, firmware development for solenoid control
Shiva Khanal: switch interrupts and logic
Gavin Philips: LCD hardware interfacing and firmware development, firmware integration with MSP libraries

Application Firmware Flowchart



Schematic



Solenoid
Solenoid
Solenoid
Solenoid
Solenoid
NPN_DARLINGTON
NPN_DARLINGTON
NPN_DARLINGTON
NPN_DARLINGTON
NPN_DARLINGTON
MSP430F5529
Level Converter
NHD-0216K1Z-FL-YBW
5V
GND