

# BAssist

(Banjo Assist Robot)

Team Members:

Fatemeh Gholizadeh

David Hatch

Shiva Khanal

Gavin Philips

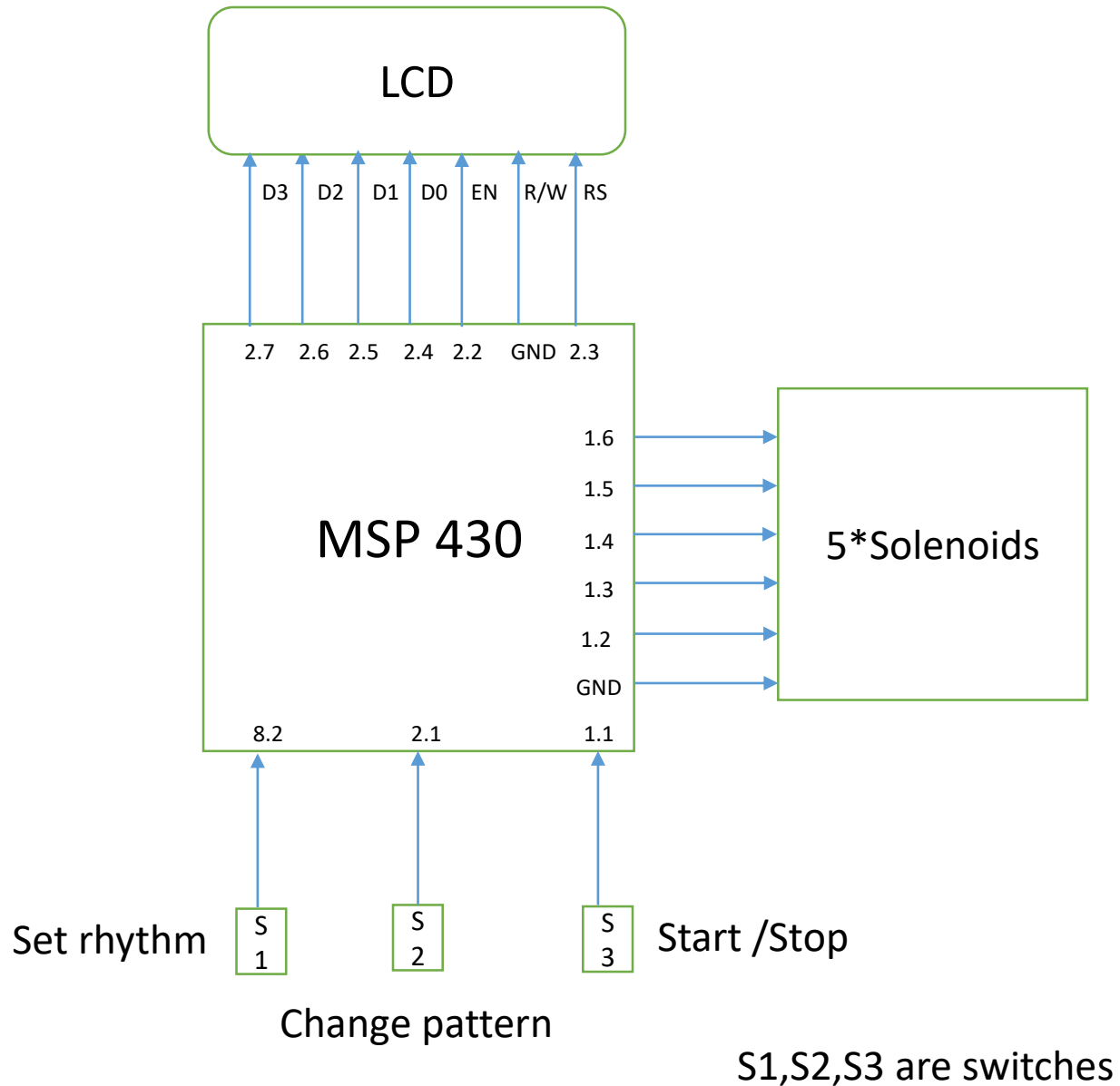
# Overview

A robotic assistant for  
Scruggs-style banjo plucking.

The BAssist will pluck the banjo strings in  
a desired pattern, leaving you free to  
focus on fretting the desired chords!



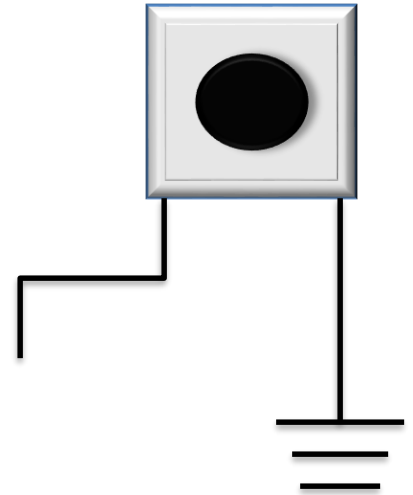
# Block Diagram



# Components

# Switches

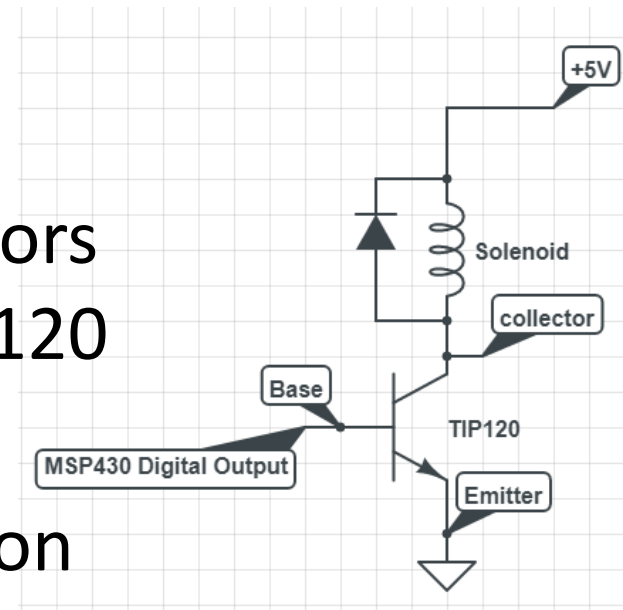
- P1.1 – Start/Stop
- P2.1 – Cycle through pattern
- P8.2 – Set rhythm
- Hardware interface: one GPIO line per switch
- Software handling: pins set up to trigger interrupt on falling edge
- Progress: interrupts implemented and tested for start/stop and pattern cycling
- Challenges: need to implement debounce





# Solenoid Actuator

- 5x - 5V mini push-pull solenoid
- Hardware interface: Solenoid actuators connected to GPIO pins through TIP120 Darlington power transistors.
- Software handling: GPIO manipulation
- Progress: plucking patterns tested and working



# LCD

- 2x16 character LCD module
- Hardware interface: 4 data lines, 3 control lines
- Software handling: LCD addressed via custom drivers that manipulate digital GPIO lines
- Progress: implemented provided pseudocode in software



# Project Status



# Task Division

- LCD and User Notifications
  - Fatemeh Gholizadeh, Gavin Philips
- Rhythm Control System
  - Shiva Khanal
- Solenoids and Mechanical Design
  - David Hatch

# Hardware

- Accomplishments
  - 3d printed prototype stabilizing housing for solenoids
  - Tested and confirmed transistor functionality
  - Tested and confirmed solenoid functionality
- Opportunities/issues
  - LCD screen acquired, but untested
  - Heat dissipation in solenoids
  - Still need to test solenoids on the actual banjo

# Software

- Accomplishments
  - Designed and implemented basic architecture
  - Implemented and tested timer peripheral @ 1kHz
  - Implemented and tested interrupts for switches
  - Implemented and tested logic that translates the plucking pattern to solenoid actuation
- Opportunities/Issues
  - LCD driver needs rigorous testing, as it is only based on timing diagrams and pseudocode provided by Newhaven Display
  - LCD driver blocks rest of application by using active wait
  - Worst case scenario, we have a fallback LCD