Wireless Midi Controller

By: Michael Brady, Sarah Palecki, Allan Belfort
Group 57
TA: Anthony Caton

Introduction

- Wireless MIDI Glove Instrument
- For DJs or artists
  - Eliminates need to stand by soundboard or computer
- Uses simple gestures
  - Flex of a finger
  - Tilt of your wrist

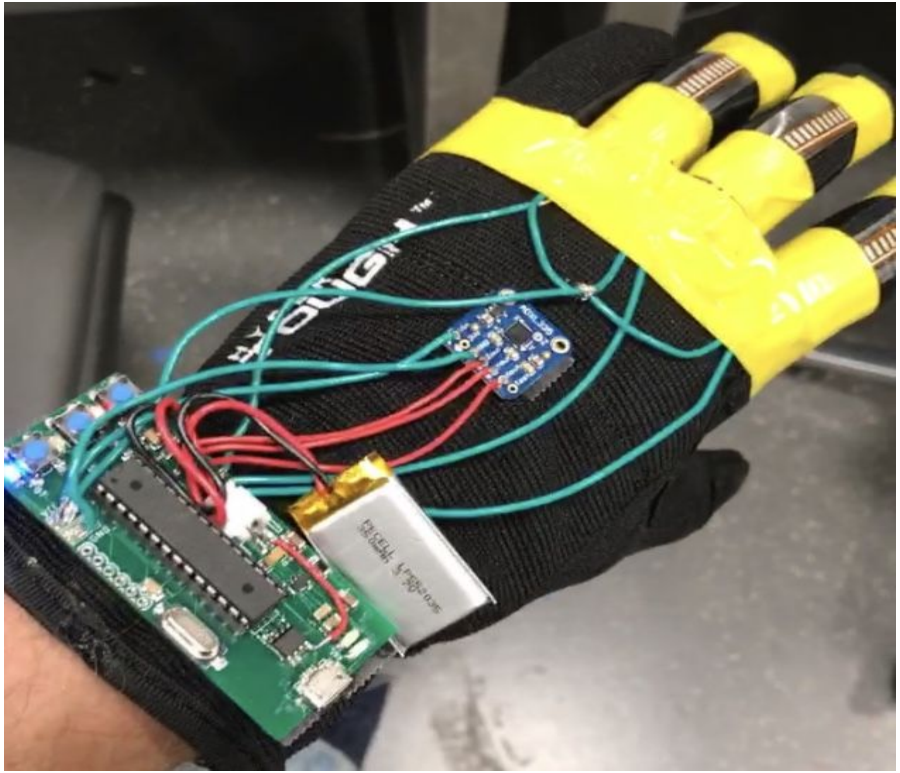
Wireless MIDI Glove

Presentation Outline

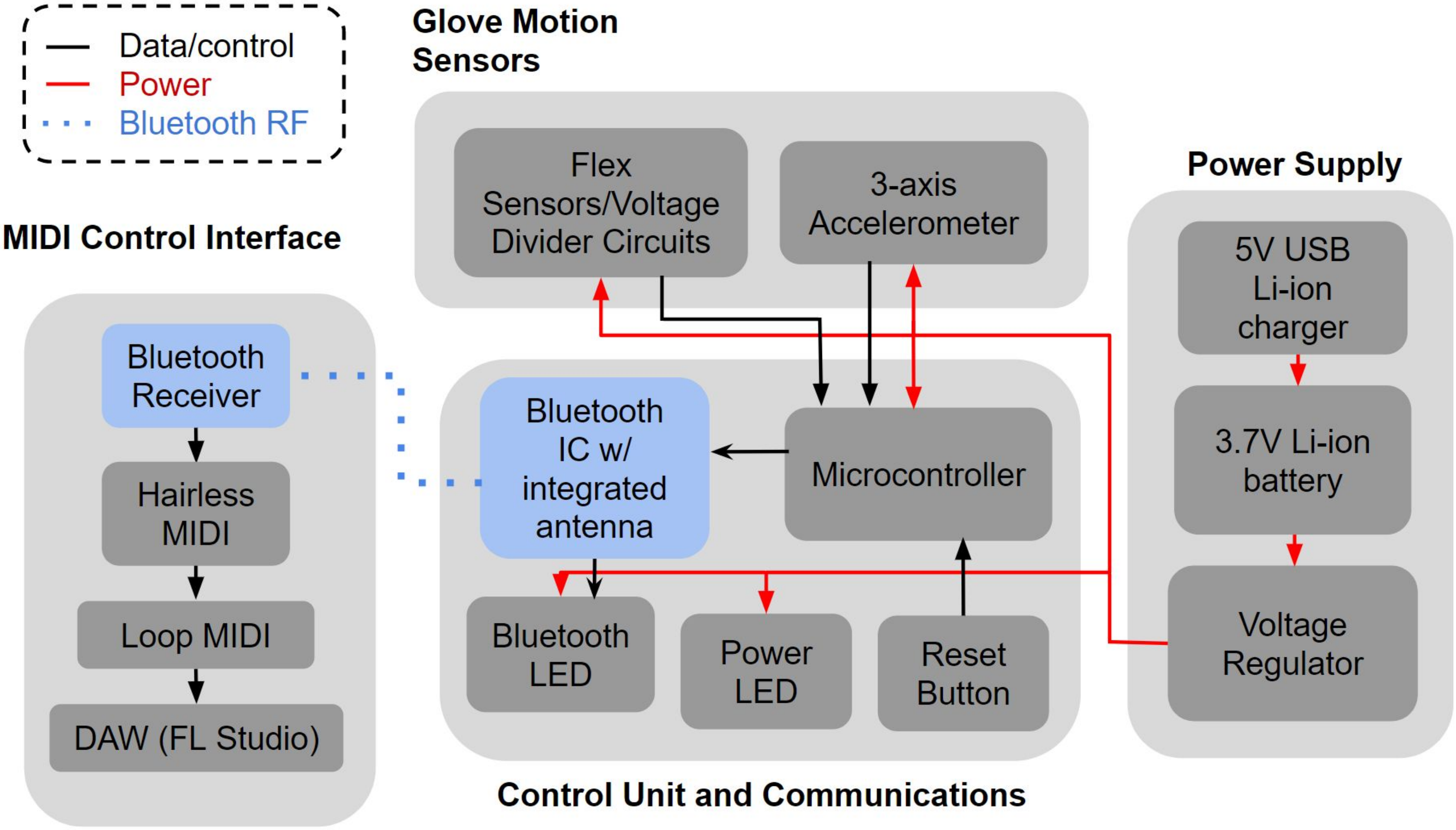
- System Overview and Block Diagram
- Hardware used
    - Sensors
    - Control Unit and Communications
    - Power Supply
- Software Used
    - MIDI Control Interface
    - MIDI Code
- Future Work/Acknowledgements/Thank You

System Overview

- Hardware:
  - Microcontroller: Atmega328
  - Bluetooth: RN-41
  - 2.2'' flex sensors x3
  - 3-axis analog accelerometer
- Software
  - Sensor mapping/MIDI outputting/Microcontroller code
  - Hairless MIDI
  - Loop MIDI
  - FL Studio



Block Diagram



Glove Motion Sensors-
3-axis accelerometer

- Adafruit Adxl335 3-axis analog accelerometer
  - Full sensing range of +/- 3G
  - Runs on 3.3V
- X-axis tilt = sound panning
- Y-axis tilt = volume control
- Output directly to the microcontroller

Glove Motion Sensors- Flex Sensors

- Sparkfun 2.2'' Flex Sensors
- 3 sensors:
  - Index, middle, ring finger
- Voltage divider circuit
- Output of voltage divider circuits are read as analog inputs to microcontroller

$$V_{out} = V_{in} \frac{(R_{flex})}{(R_{flex} + R_2)}$$

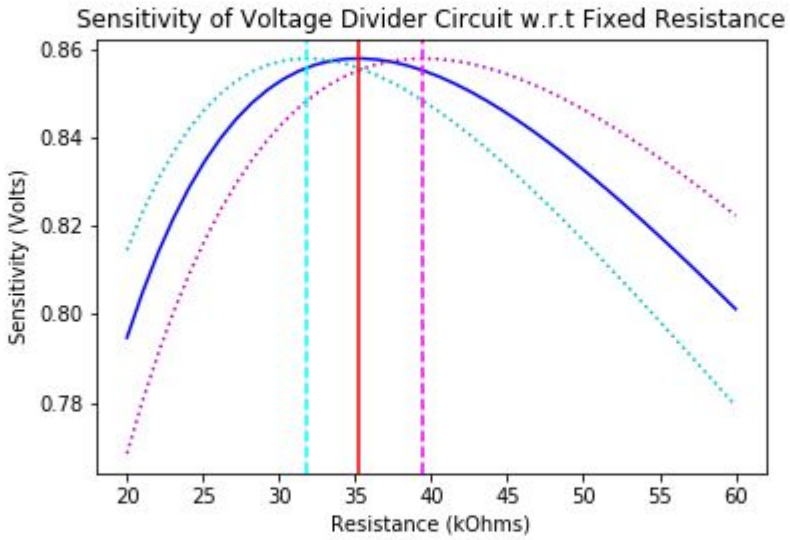
Analog Sensor Mapping

$$Output = \frac{Max - Input}{Max - Min} * 127$$

pppp	= 8
ppp	= 20
pp	= 31
p	= 42
mp	= 53
mf	= 64
f	= 80
ff	= 96
fff	= 112
ffff	= 127

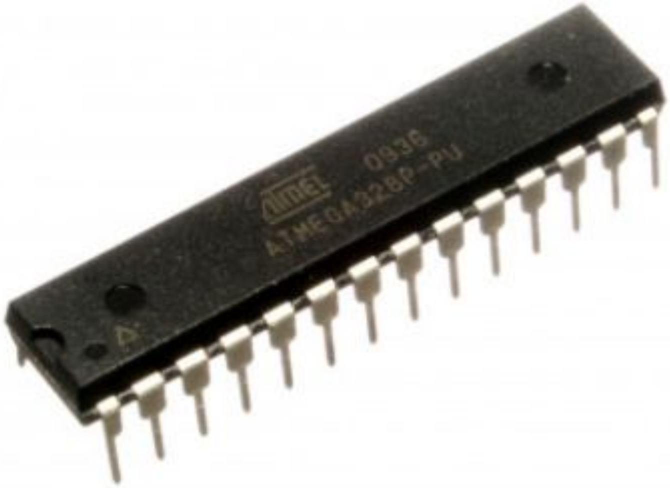
Flex Sensors Voltage Divider Data and Tolerance Analysis

- Maximize sensitivity of voltage divider circuit
- Looked for fixed resistance
- Modeled the errors
  - Due to inconsistencies in sensors

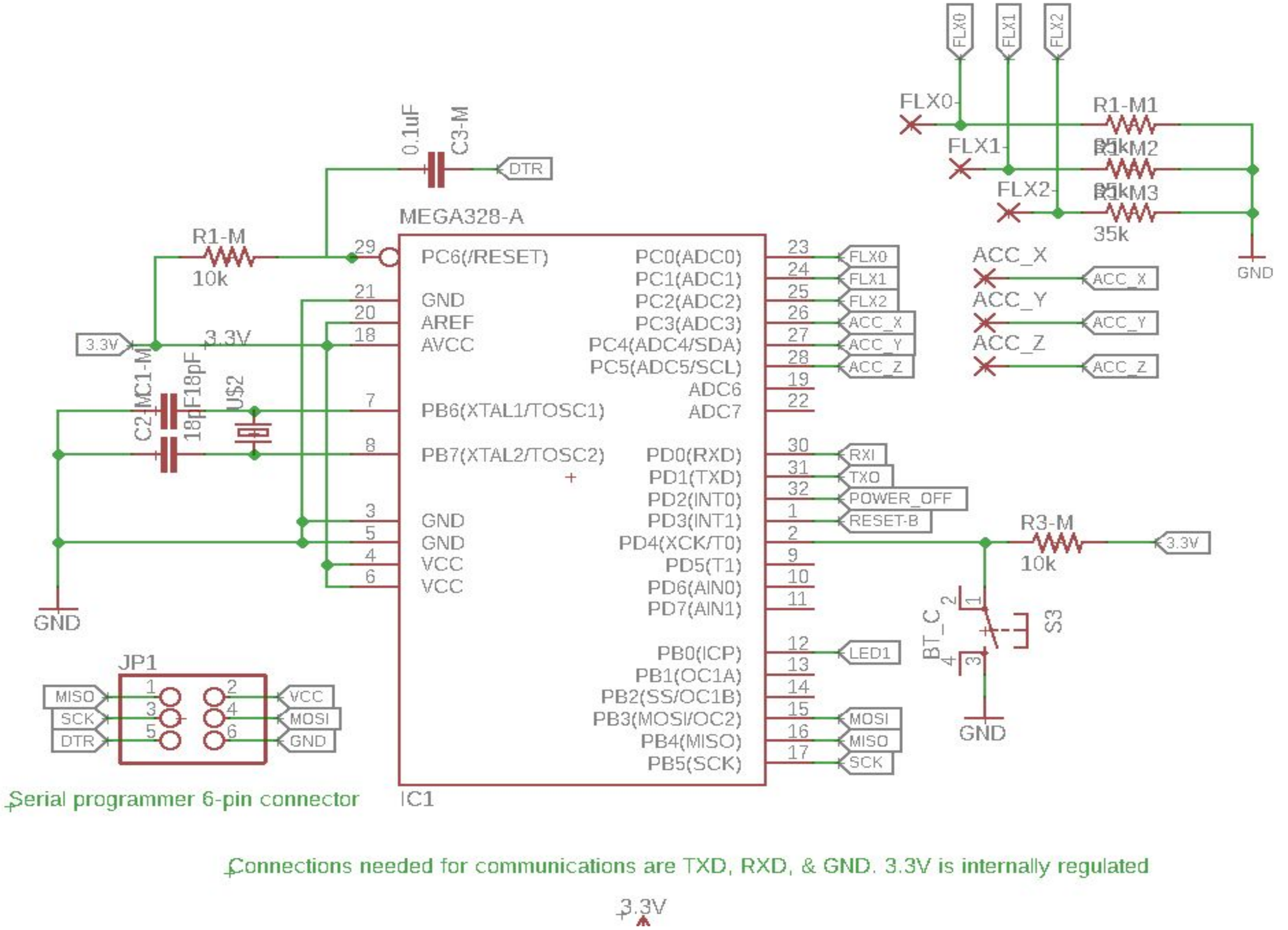


Control Unit and Communications- Microcontroller

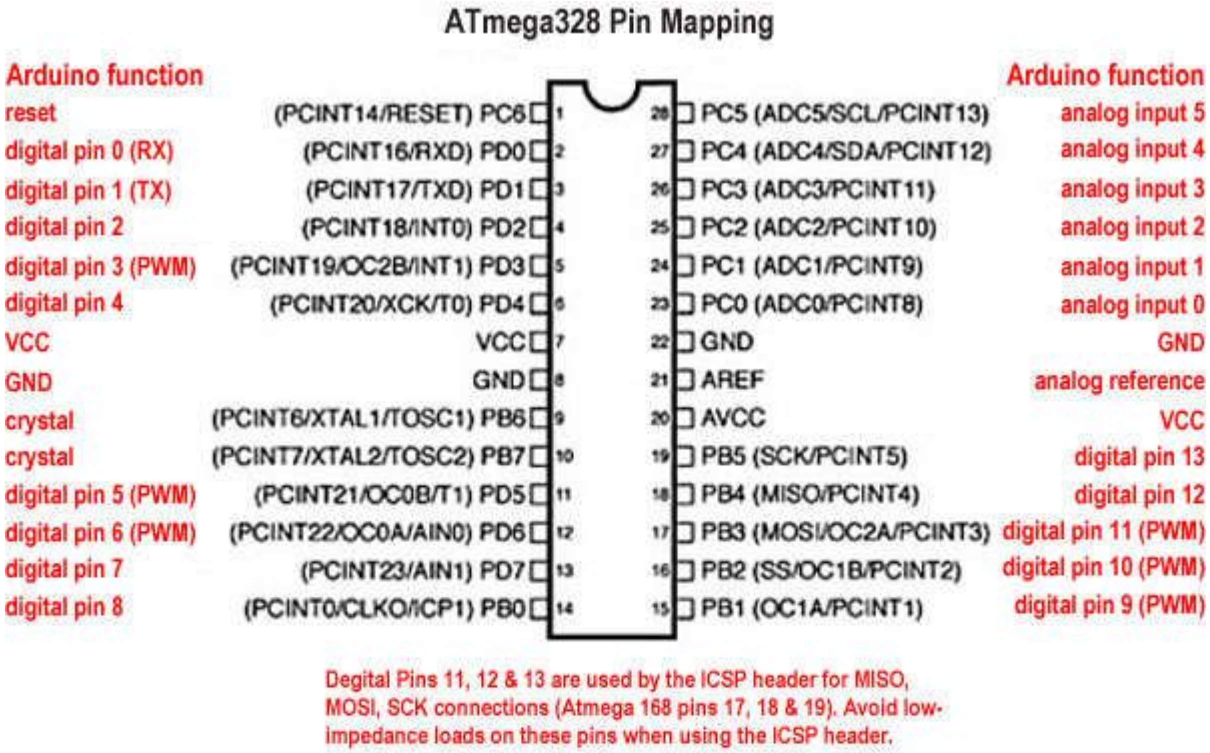
- Atmega328p
    - 8-bit AVR RISC based
    - 32KB ISP Flash memory
    - Operating at 3.3V
    - 8 MHz clock



Microcontroller: Atmega328 Schematic



Microcontroller: Programming (Pin map to Arduino map)

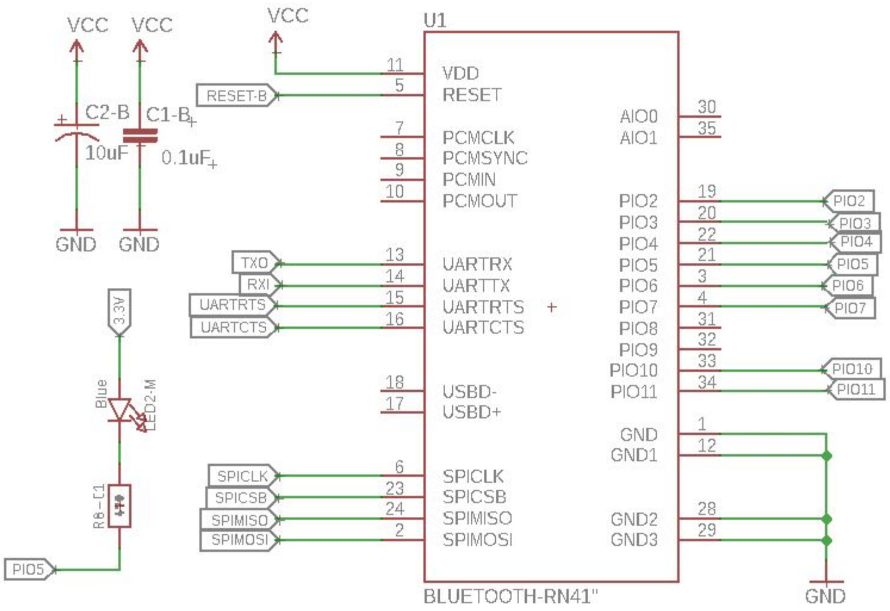


Control Unit and Communications: Bluetooth Transceiver

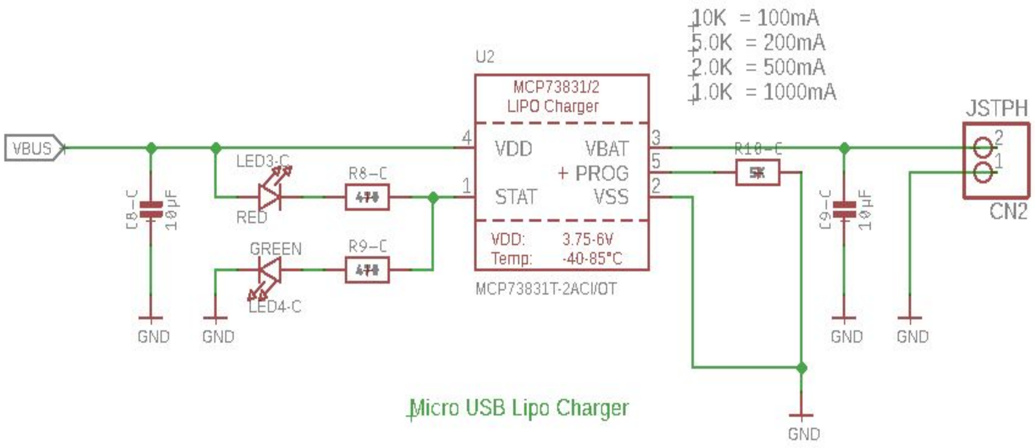
- Model RN-41
  - Fully certified class 1 Bluetooth
  - Used in transmitting mode
  - 115,200 baud
  - Bluetooth receiver is built in to computer or added via bluetooth-usb receiver
  - Blue LED for debugging

Bluetooth Transceiver: Programming

- Necessary to pair the transmitting RN-41 Bluetooth module with the Bluetooth module located inside a laptop
- Easy first-time setup through computer
- LED on PIO5 (Status) pin to show connection state



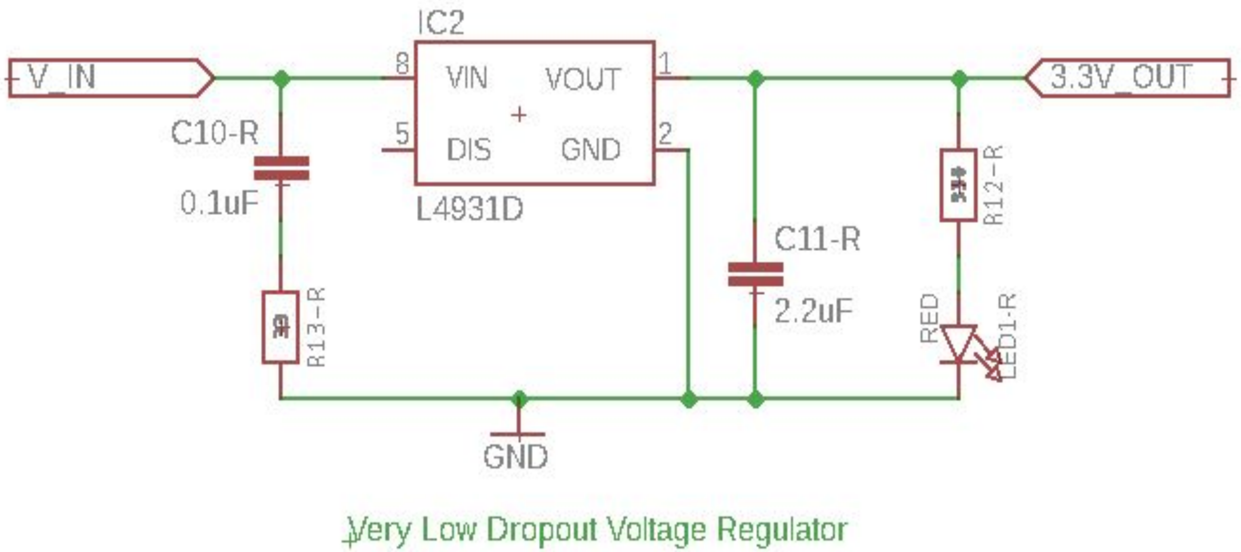
Power Supply: Li-Ion Battery & USB Charger



- MCP73831T charging IC
  - 5V micro USB input
  - Variable charge rate
    - Optimal charging speed
    - Extended battery life
  - Full charge in 1-2 hours
  - Charge status LEDs
    - Red- charging
    - Green- charge complete
- 350 mAh Li-Ion battery
  - 3.7-4.2V output depending on charge
  - Device run time > 6 hours

Power Supply: Voltage Regulator

- L4931 very low dropout regulator
  - Input 3.7-4.2V from battery
  - Output 3.3V +/- 1%
  - 4-6 mA current draw

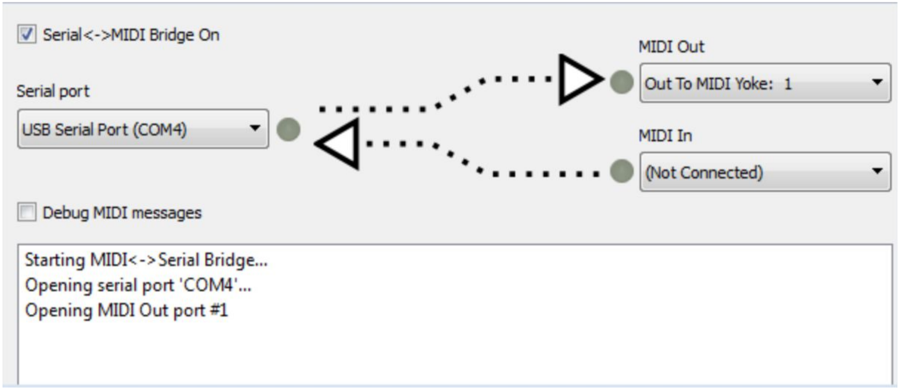


MIDI Control Interface: Bluetooth Receiver

- Located inside of most laptops
  - External bluetooth-USB receiver can be used if needed
- Simple pairing procedure
- We used the software “Tera Term” to debug the Bluetooth output
- Settings used:
  - Baud rate of 115200 bits/second
  - Line ending: CR-LF

MIDI Control Interface: Hairless MIDI

- Serial <-> MIDI Bridge
- Debug tool for MIDI messages

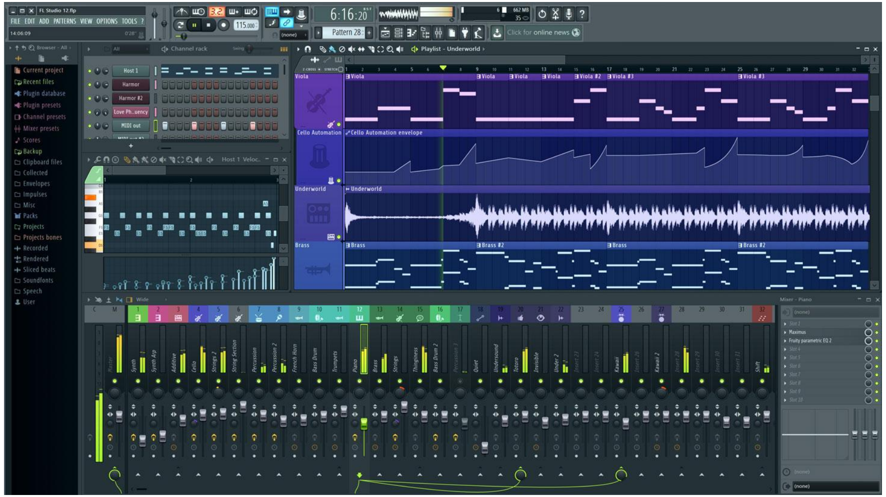


MIDI Control Interface: Loop MIDI

- Create virtual MIDI ports on the computer
  - Free
  - Easy to download
- Input from Hairless MIDI
- Output to DAW
  - Lets DAW recognize data as MIDI

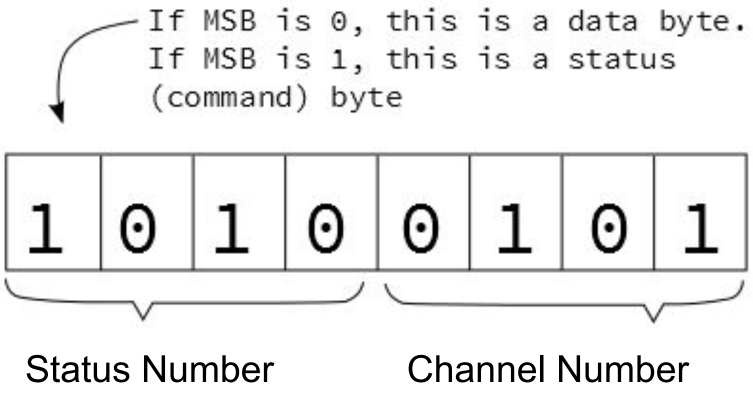
MIDI Control Interface: DAW

- FLStudio
- Allows musicians to create, edit, record or arrange their music
- Works with MIDI protocol to record and play MIDI files
- Received the glove's codes and processed them to create/modify sounds



MIDI Control Interface: MIDI Code

- Used the Arduino Midi Library by FortySevenEffects on Github
  - Allows for use of Midi.SendNoteOn(note, channel, velocity);
    - Cleans up the code
- MIDI Code Format
  - Status Bytes
  - Data Bytes
- Two types of MIDI messages:
  - “Note On”
  - “Control Change”
  - Made up of Status + Data Bytes



MIDI "Note On"

- A "Note On" Midi message contains 3 bytes:
  - 1 "Status Byte": 0x9?
    - ? = Channel
  - 2 "Data Bytes": 0x?? + 0x??
    - 1st Byte ?? = note value (60 = middle C)
    - 2nd Byte ?? = velocity value
- Velocity value is used to change volume

MIDI Status Messages					
Message Type	**MS Nybble***	**LS Nybble***	**Number of Data Bytes**	**Data Byte 1**	**Data Byte 2**
Note On	0x9	Channel	2	Note Number	Velocity

* A Nybble is equivalent to 4 bits

MIDI "Control Change"

- A "Control Change" Midi message contains 2 bytes:
  - 1 "Status Byte": 0xB?
    - ? = Channel
  - 1 "Data Byte": 0x??
    - ?? = Controller Number
- Used to change effects
  - Pan
  - Vibrato
  - Pitch Bend

MIDI Status Messages				
Message Type	MS Nybble*	LS Nybble*	Number of Data Bytes	Data Byte 1
Control Change	0xB	Channel	1	Control Number

* A Nybble is equivalent to 4 bits

Future Hardware Development

- Power Button
  - Turn device On/Off
- LEDs
  - In fingers with flex sensors
  - Change intensity
- Pressure sensors
  - In fingertips
  - Control different sounds
- Better materials
- Slimmed down package

Future Software Development

- Enable a “tapping” feature on the accelerometer
  - Uses sudden acceleration along z-axis
  - “Single tap” vs. “Double tap”
  - Change programs
  - Control separate functions
- Change the sensor mapping algorithm
  - Simple linear algorithm vs. specifically designed algorithm for each sensor

Overall Analysis of Wireless MIDI Glove

Strengths:

- Can be used at a range of ~50 feet from computer
- Low-latency

Weaknesses:

- Possible difficulty in user setup
- Only produces sounds, does not edit them

Threats:

- Global DJ Tornado Gloves
- Remidi Glove and Controller

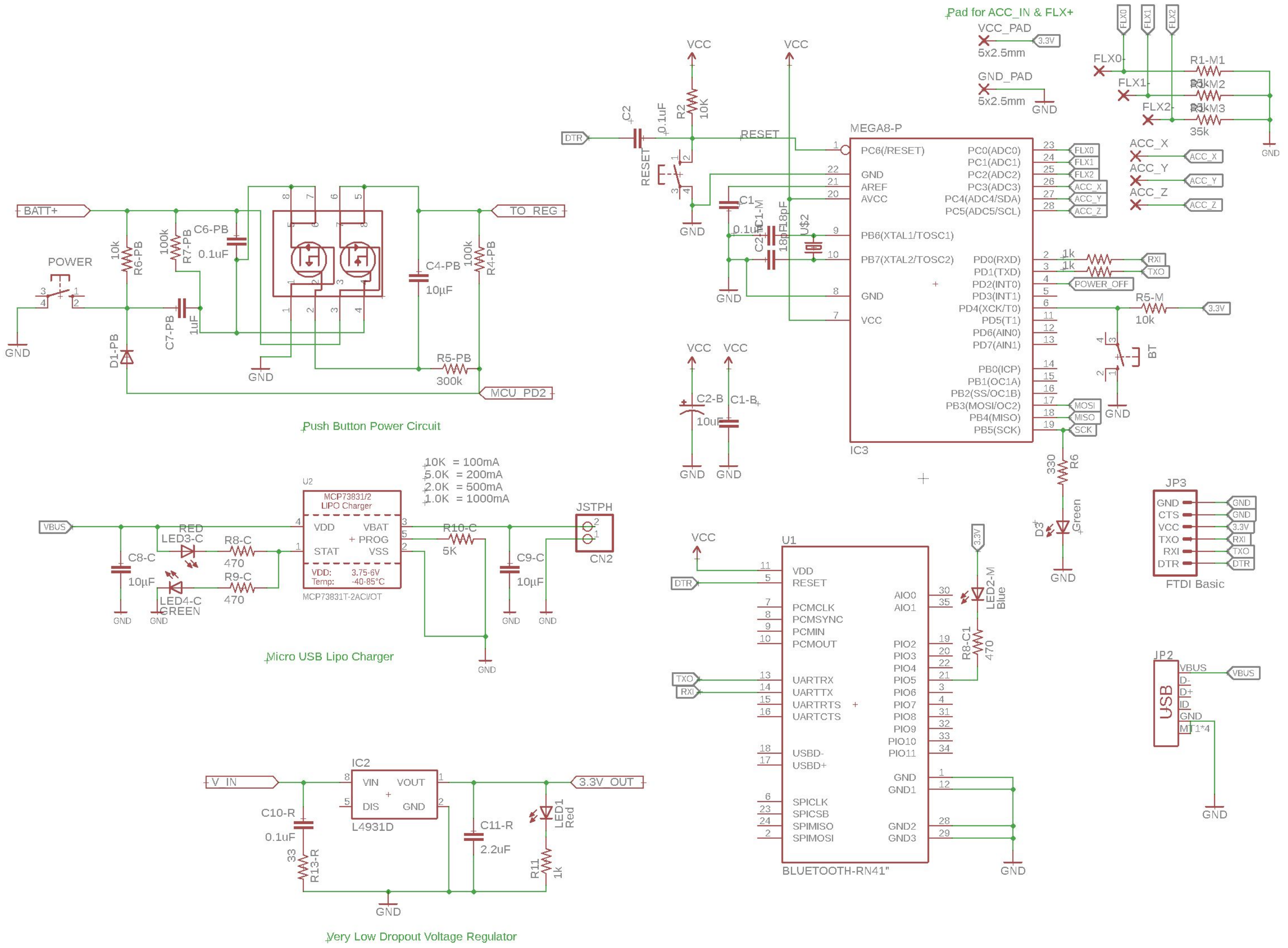
Opportunities:

- Innovative performance wear
- Can use in addition to supplement a different instrument

Acknowledgements

- Professor Lippold Haken, ECE 402
- Mr. Anthony Caton

Thank you!



Pad for ACC_IN & FLX+
VCC_PAD
5x2.5mm
GND_PAD
5x2.5mm
MEGA8-P
IC3
PC6(/RESET)
GND
AREF
AVCC
PB6(XTAL1/TOSC1)
PB7(XTAL2/TOSC2)
VCC
PC0(ADC0)
PC1(ADC1)
PC2(ADC2)
PC3(ADC3)
PC4(ADC4/SDA)
PC5(ADC5/SCL)
PD0(RXD)
PD1(TXD)
PD2(INT0)
PD3(INT1)
PD4(XCK/T0)
PD5(T1)
PD6(AIN0)
PD7(AIN1)
PB0(ICP)
PB1(OC1A)
PB2(SS/OC1B)
PB3(MOSI/OC2)
PB4(MISO)
PB5(SCK)
POWER_OFF
BATT+
TO_REG
MCU_PD2
POWER
Push Button Power Circuit
MCP73831/2
LIPO Charger
VDD
VBAT
STAT
PROG
VSS
VDD: 3.75-6V
Temp: -40-85°C
MCP73831T-2ACI/OT
10K = 100mA
5.0K = 200mA
2.0K = 500mA
1.0K = 1000mA
JSTPH
CN2
Micro USB Lipo Charger
IC2
VIN
VOUT
DIS
GND
L4931D
V_IN
3.3V_OUT
Very Low Dropout Voltage Regulator
U1
BLUETOOTH-RN41"
VDD
RESET
PCMCLK
PCMSYNC
PCMIN
PCMOUT
UARTRX
UARTTX
UARTRTS
UARTCTS
USBD-
USBD+
SPICLK
SPICSB
SPIMISO
SPIMOSI
AIO0
AIO1
PIO2
PIO3
PIO4
PIO5
PIO6
PIO7
PIO8
PIO9
PIO10
PIO11
GND
GND1
GND2
GND3
JP3
GND
CTS
VCC
TXO
RXI
DTR
FTDI Basic
JP2
USB
VBUS
D-
D+
ID
GND
MT1*4