Wind Instrument Arduino MIDI-Controller

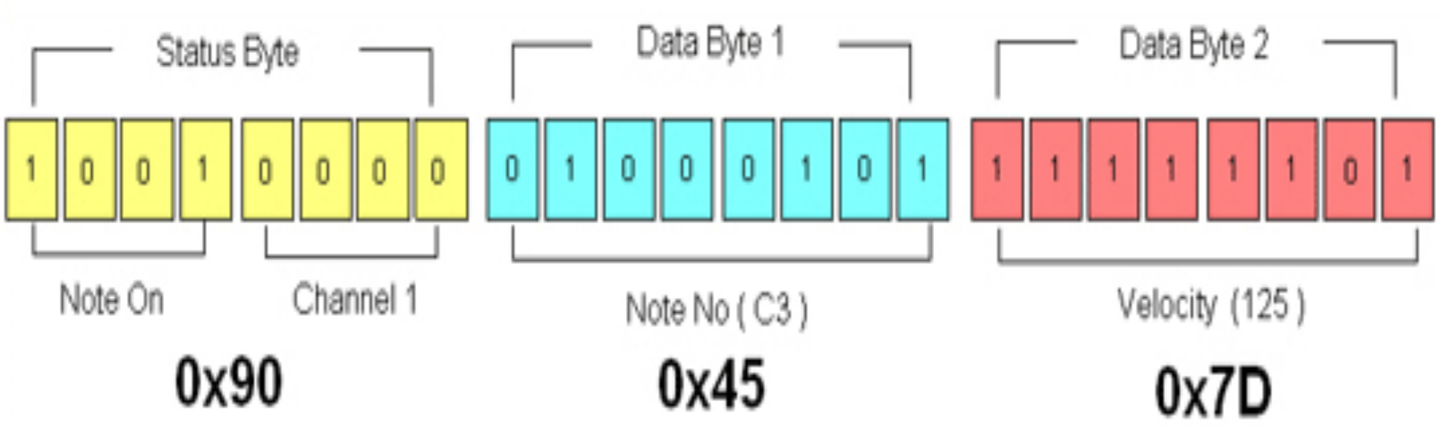
Pablo Sanchez 146691
Albert Valls 146755

Overview

1. What is an Arduino MIDI-controller? (video)
2. Motivation
3. Our goal
4. Which materials and sensors were necessary?
5. Final Circuit Schematic
6. Project code
7. Results
8. Conclusions
9. References

Arduino MIDI-Controller

- What is MIDI?
  - Musical Instrument Digital Interface
  - Two type of bytes: Command and data



Motivation

- Why build a MIDI-controller?

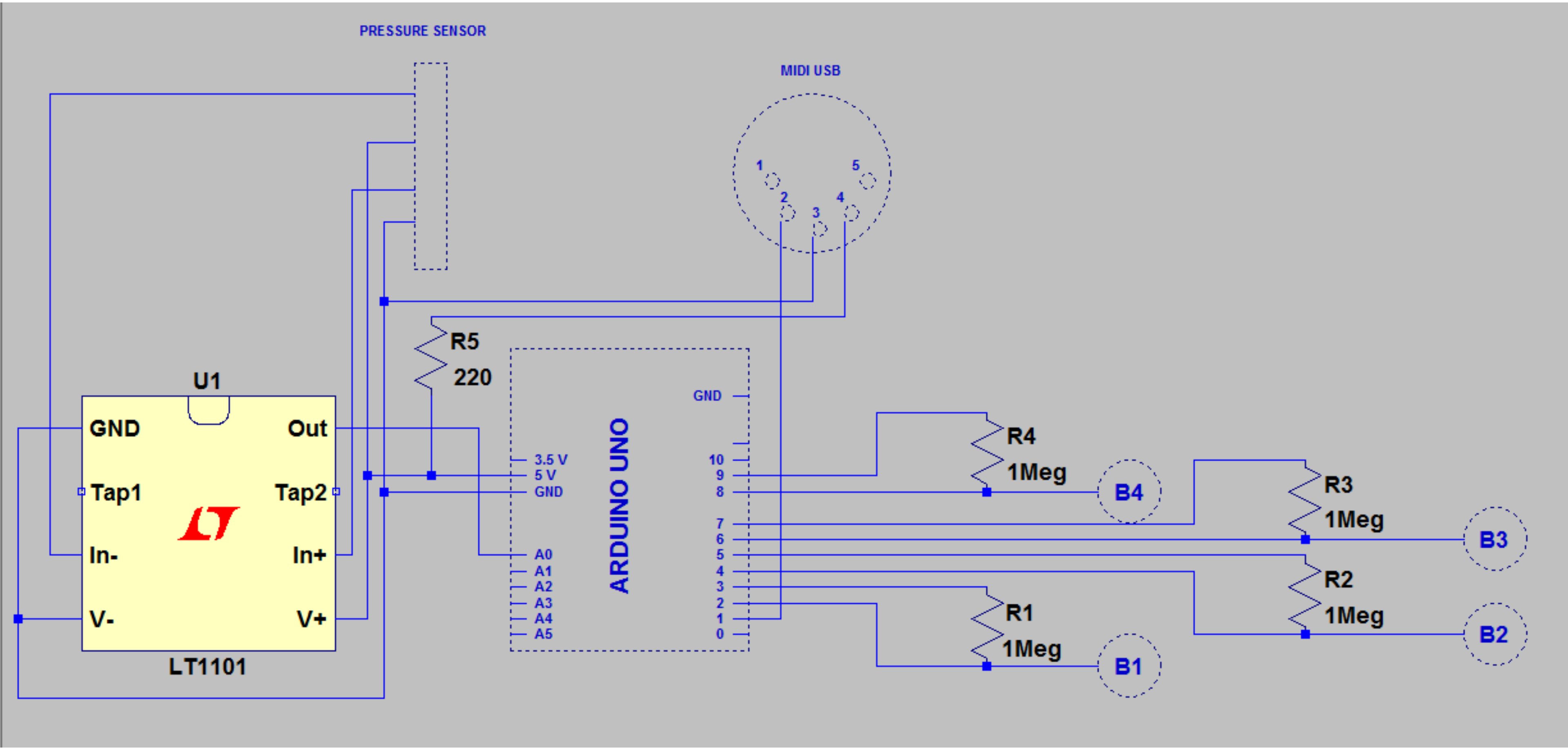
Our goals

- Successfully build the intended circuit
  - Touch sensitive sensors
  - Pressure sensor
  - MIDI connectors
- If possible, successfully implement the wind instrument MIDI-controller

Which sensors and materials were necessary?

- Arduino UNO board
- ProtoShield
- Hook-up Wire
- Pressure sensor MPX5010GP
- One LT1101 Op-amplifier
- Four Aluminum Wire
- Four 1M Ω Resistors
- One MIDI-USB cable
- One 220 Ω Resistor
- One female MIDI connector

Final Circuit Schematic



Project Code

- Pressure sensor

```
 int sensorValue = analogRead(A0);
  double outputValue = map(sensorValue, 0, 1023, 0, 127);
  double parentesis = outputValue / 25;
  double velocity  = (20*80)*(log10(parentesis));

  if ( outputValue >25){
    sensor_active = true;
    delay(100);        // delay in between reads for stability
} else {
        sensor_active = false;
}
```

Project Code

- Touch sensitive sensors

```
 if ((val1 == HIGH) && (val2==HIGH) &&
(val3==HIGH) && (val4==HIGH)){
    notes [0] = true ;     }
      while ((val1 != HIGH) || (val2 !=HIGH) ||
(val3 !=HIGH) || (val4 !=HIGH)) {
       delay ( 100);
       if ((val1 != HIGH) && (val2 == HIGH) &&
(val3 == HIGH) && (val4 == HIGH)){
       cap1++;
       notes [1] = true ;
       nota_actual = 0;
       }else if ((val1 == HIGH) && (val2 != HIGH)
&& (val3 == HIGH) && (val4 == HIGH)){
       cap2++;
       notes [2] = true ;
       nota_actual = 1;
       }else if...
```

```
    val1 = digitalRead(3);    // re-read the input to
be checked
       val2 = digitalRead(5);    // re-read the input
to be checked
       val3 = digitalRead(7);    // re-read the input
to be checked
       val4 = digitalRead(9);    // re-read the input
to be checked
```

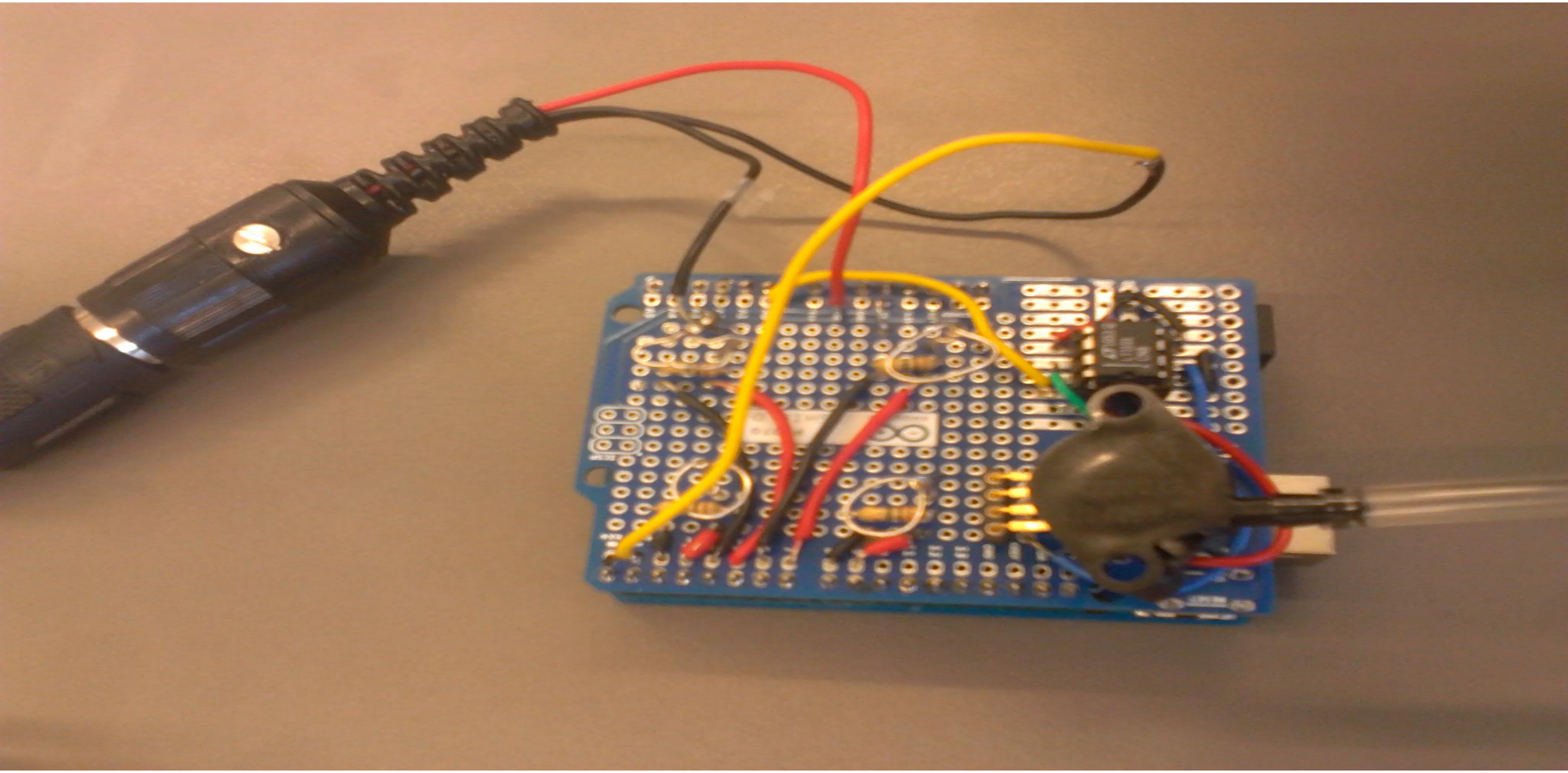
Project Code

- MIDI connectors

```
int primeraoctava [12] = {0x3C, 0x3D, 0x3E, 0x3F, 0x40, 0x41, 0x42, 0x43, 0x44, 0x45, 0x46, 0x47};
int segonaoctava [12] = {0x48, 0x49, 0x4A, 0x4B, 0x4C, 0x4D, 0x4E, 0x4F, 0x50, 0x51, 0x52, 0x53};
int terceraoctava [12] = {0x54, 0x55, 0x56, 0x57, 0x58, 0x59, 0x5A, 0x5B, 0x5C, 0x5D, 0x5E, 0x5F};
  void loop(){
 if( first_active == true){
 int note1 = primeraoctava[nota_actual];
    noteOn(60, note1, velocity);
    delay(300);
    noteOn(60, note1, 0);
    delay(200);
  }
}
void noteOn(int cmd, int pitch, int velocity) {
  Serial.write(cmd);
  Serial.write(pitch);
  Serial.write(velocity);}
```

Results

- Successfully built the intended circuit



Results

- Unsuccessfully implemented the Wind instrument MIDI-Controller

Conclusions

- Difficult to successfully implement a Wind Instrument MIDI-Controller via Arduino UNO.
- Complications with our code and the expected results.
- Overall glad with what we achieved, but it would have been good to achieve both goals successfully.

References

- [1]http://arduino.cc/en/Tutorial/Midi
- [2]http://www.dtic.upf.edu/~jlozano/interfaces/interfaces7.html
- [3]http://playground.arduino.cc//Main/CapacitiveSensor?from=Main.CapSense
- [4]http://www.instructables.com/id/Send-and-Receive-MIDI-with-Arduino/
- [5]http://itp.nyu.edu/physcomp/Labs/MIDIOutput