# Piano Playing Arm
# Group 7

Dembski, Clayton John
Rangel, Steven
van Rossum, Floris
White, Adam

# Background / Objectives

- Develop the hardware and software necessary to enable an ABB Industrial Arm to play a musical keyboard
- Decide on method for the robot to physical press the keys
  - Moving the arm
  - Hydraulic actuator
  - Electrical solenoid
- Write RAPID code where a user can enter a song and the software will interpret the notes into movements

# Hardware

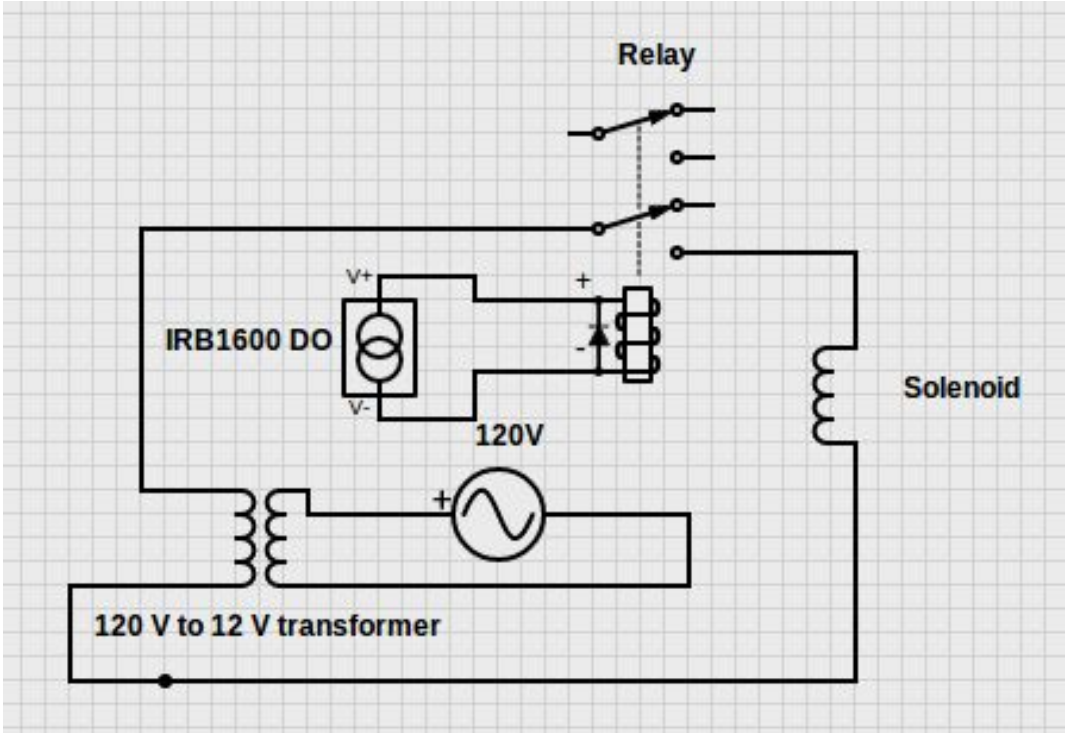
- A Casio SA-76 Keyboard was used
- 12V DC, 2.5 A, Electric solenoid
- 24V relay to actuate the solenoid from robot controller
- 12 V Power Supply



# End Effector

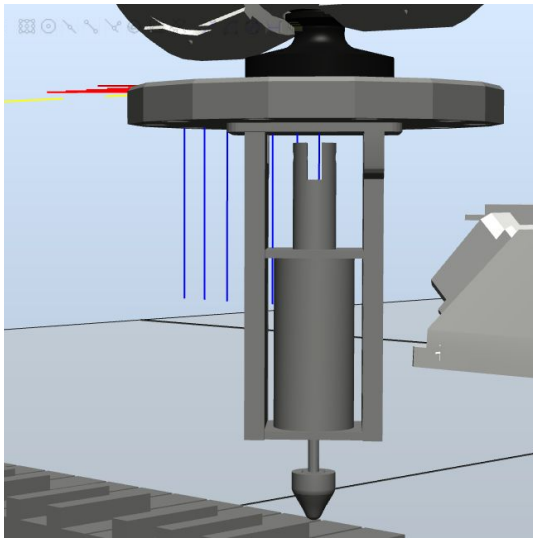

- Solenoid pushes with a large amount of force
- Bouncy ball to prevent damage and noise

# Relay Switch Circuit



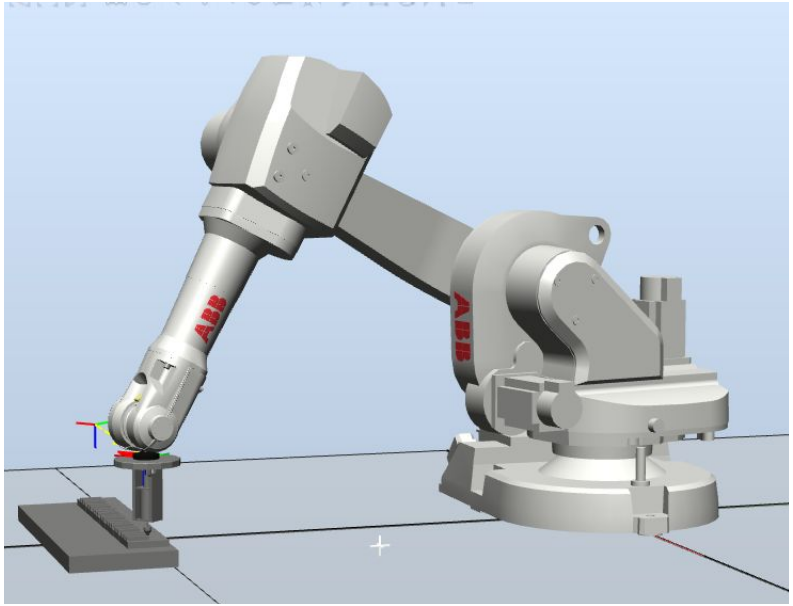
# Robotstudio Simulation

- Used CAD Model of the end effector and the keyboard
- Establish working configurations of the arm
- Establish timing of moves
- Establish waiting protocol



# RAPID Code

- Moving between keys
  - MoveL commands
- Timing
  - WaitTime
  - ITimer
- Push
- Retract



# GoToKey(num keyNum) Algorithm

```
PROC GoToKey(num keyNum)
    !Some delicious RAPID code that may or may not do things of important nature to do with the robotics


    ! Do some calculations to determine the location of the position
    ! X: Variable
    ! Y: -1183.2 mm
    ! Z: 125.1 mm

    VAR num distWW := 20.31;
    VAR num distBW := 22.78/2;
    VAR num distance := 0;

    IF keyNum MOD 12 > 0 and keyNum MOD 12 < 7 THEN
        distance := distBW * (keyNum MOD 12);
    ENDIF

    IF keyNum MOD 12 = 7 THEN
        distance := distBW * 6 + 2*distBW;
    ENDIF

    IF keyNum MOD 12 > 7 AND keyNum MOD 12 < 12 THEN
        distance := distBW * 6 + 2*distBW + ((keyNum MOD 12) MOD 7) * distBW;
    ENDIF

    distance := distance + Trunc(keyNum / 12) * (10 * distBW + 4 * distBW);
    TPWrite "Distance calculated: " \num:=distance;
    MoveL [[-distance + 26.0,-1183.2,125.1],[0.55982,0.38937,-0.58583,0.43795],[-2,-1,2,1],[9E+09,9E+09,9E
ENDPROC
```

# GoToKey(num keyNum) Hardcoded

```
PROC GoToKey(num keyNum)
    timecount := 0;
    !SetDO D652_10_DO1, 1;
    !Some delicious RAPID code that may or may not do things of important nature to do with the robotics

    ! Do some calculations to determine the location of the position
    ! X: Variable
    ! Y: -1183.2 mm
    ! Z: 125.1 mm

    ! Retract the pusher

    ! Wait a bit
    IF keyNum = -1 THEN
        WaitTime(5 - timecount * 0.1);
    ELSE
        Retract;
        IF keyNum = 1 THEN
            MoveL [[26.79,-1183.27,140.6],[0.55992,0.389101,-0.585975,0.437862],[-1,0,1,1],[9E+09,9E+09,9E+0
```

# GoToKey(num keyNum)

- Moving from one key to the next
- Hard coded version with online programming
- Input parameter 1 - 44
  - Each representing a specific key
- -1 is a hold

# Timing

- The biggest challenge
- Using wait commands and interrupts in order to press keys consistently
- Interrupts
  - Managed the timing between keys
- Wait commands
  - Managed timing for retracts and pushes
  - Compensated for a short move

# Retract and Push

- Toggling D652_10_DO1 to 1 or 0, retracted or extended the solenoid
- Wait commands were essential
  - Prevents dragging tool over keyboard
  - Allows tool to fully extend
- Wait commands had to be properly synced up with interrupt for program to function

# Final Setup


# Demo

# Future Improvements

- Increase solenoid switching speed
- Play Chords
- Easier Calibration
- Reduce mechanical noise
- Read Midi Data

# Division of Labor

| | |
|---|---|
| Dembski, Clayton John | RAPID code, Demo Setup |
| Rangel, Steven | Hardware, CAD |
| van Rossum, Floris | Robot Studio, Demo Setup |
| White, Adam | Hardware, Electrical |

# Questions?