



The TI-Innovator Piano

Mini-Project 5: Play the Piano Using the TI-Innovator Rover

In this last mini-project, you will import the piano code from the brightness sensor. You will use the ultrasonic sensor and the TI-Innovator Hub to measure distances to the sensor. Based on the distance, draw a circle on the correct key and play the corresponding note.

Objectives:

- import PIANOBRT to draw your keyboard and import the loop
- use the range sensor on TI-Rover
- use IF statements to make decisions
- use LISTS to store information
- use a WHILE statement to repeat code

The Piano Project Overview:

After completing a series of 5 mini-projects, you will have several ways to play a piano. You will be able to play the piano using the keys on your handheld, using the brightness sensor on the TI-Innovator Hub, using a separate ultrasonic sensor or using the ultrasonic sensor on the TI-Rover.

Mini-project order:

1. Piano setup
2. Play the piano using the keypress event.
3. Play the piano using the brightness sensor on the TI-Innovator Hub
4. Play the piano using an ultrasonic sensor an the TI-Innovator Hub
5. **Play the piano using the ultrasonic sensor on the TI-Innovator Rover**



Tech Tip:

This activity can be completed without doing mini-project 4. This is an alternative way to using an external sensor, depending on material availability.

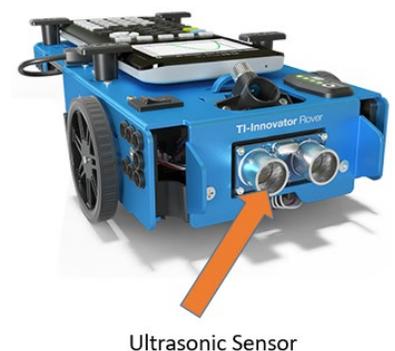
1. Let's make sure your TI-Rover is working. Create a temporary project named T.
Type the following lines of code
You will get most of the code in the ROVER menu located in the HUB menu.

```

:Send("CONNECT RV")
:getKey →K
:While K≠45
:getKey →K
:Send("READ RV.RANGER")
:Get(D)
:Output(6,1,D)
:End

```

Execute your code. Hover your hand over the sensor. What types of values do you get?



Ultrasonic Sensor



10 MOC: Beyond Basics

TI-84 PLUS CE TECHNOLOGY

THE TI-INNOVATOR PIANO: MINI-PROJECT 5

STUDENT ACTIVITY

5. If your ranger is reading in meters, most of your numbers in the temporary project were under 1. If you multiply the distance by 100, your distance values will be about the same as the brightness variables from project 3. Therefore we can use the following code

```
:0→K  
:Send("CONNECT RV")  
:While K≠45  
:getKey →K  
:Send("READ RV.RANGER")  
:Get(D)  
:iPart(D*100/7.14) →N  
:Send("SET SOUND eval(261.14*2^(N/12))")
```

