

Circuit Playground Express

*Networking – The Chicken Detector*

This lesson introduces networking, which is the ability of two or more computers to communicate with each other. The Circuit Playground supports networking via an infrared signal, the same technology used in a television remote control. Makecode supports networking with the two blocks that can be found in the “Network” category.

We will go through a few steps, but the final project in this lesson will be a “chicken detector” – inspired by one of our team members’ problem of unwanted chickens coming up the stairs to make a mess on the deck. This application allows a Circuit Playground in one place to detect a chicken passing near it with the Proximity Detector, and to send a signal to another that triggers an alarm to sound. This basic idea—a sensor attached to one Circuit Playground that causes a signal to be sent to another—can be adapted to any number of fun applications.

The Basics

Let’s start by creating a very simple networking application: two Circuit Playgrounds, programmed so that the lights on one of them respond to button presses on the other.

You will need two different Makecode programs for this application. One will run on the Circuit Playground that sends the infrared signals, and the other will run on the Circuit Playground that receives them.

The basic form of a Makecode program that uses networking is: One uses the “send number X” block to send a signal, and the other (or others—you can have any number of receivers respond to a signal) uses a “when signal received” block, usually with a “if number = X” conditional block to respond only to certain numbers.

*Tip: If you have a classroom with many groups working, it would be a good idea to assign different numbers for each team to use. That will prevent you from having one team’s signal trigger another team’s receiver. The “when signal received” block is just listening for a number; it has no way of knowing which Circuit Playgroun sent it.*

*Basic Program: Press a button on one Circuit Playground to change the color of the lights on another.*

A screen shot of a smart phone

Description automatically generated A picture containing grass, monitor, holding, sitting

Description automatically generated

Basic program: Code for the sender

Basic program: Code for the receiver

*Note: in the sender code, the “set pixel color” blocks are not really necessary. As in previous lessons, we use lights to make it a bit easier to debug the program.*

Try out your program: Load the sender code onto one Circuit Playground and your receiver code onto another. If both are running, you should be able to see that pressing the A or B button on the sender causes the lights on the receiver to change color. Experiment with the infrared communication; how far does it work? Are there angles that work better? Does it work through an obstacle?

Now that we have a basic understanding of how one Circuit Playground can communicate with another, let’s build something a bit more complex. I have four chickens living in my back yard, and they like to come up on the deck, where they make a horrible mess. I decided it would be handy if a sensor on the stairs could detect the chickens coming onto the deck and make an alarm inside my house go off, so that I could go out and chase the chickens away.

This project combines the networking code that we used above with the ultrasonic sensor that we’ve already used in previous lessons. First, let’s set up the sensor, which will send out a signal if it detects a chicken walking by.

A close up of a device

Description automatically generated

Wiring schematic for the sensor half of the Chicken Detector.

A screenshot of a cell phone

Description automatically generated

Code for the sensor half of the Chicken Detector

Just like the Basic Program we started with, we use lights on the Circuit Playground to make debugging easier. If we see the blue light light up, we know that the Circuit Playground is also sending our infrared signal. Otherwise, if your Chicken Detector is not working, it is very difficult to diagnose whether it is the sender or the receiver that is the source of the problem.

Test the sensor and make sure your Circuit Playground is lighting up as expected when something crosses in front of it. If you don’t have a chicken handy, or your chickens are not cooperating, a small child can be used as a substitute.

Once the sensor half of the Chicken Detector is working, it’s time to add the receiver. The receiver is very simple. All it has to do is listen for the infrared signal from the sender, and play a sound when the signal arrives:

A screenshot of a cell phone

Description automatically generated

Code for the receiver half of the Chicken Detector

Of course, this is just a starting point. You can put any code in the receiver at all. For example, you could wire up a servo motor from one of our previous lessons, and have the receiver wave a flag or open or close a door when it receives a signal. You could have the receiver turn an electric motor on and off to drive the wheels of a toy car, turning your Circuit Playgrounds into a remote-controlled car. You could add additional receiver Circuit Playgrounds that each perform different actions when different signals are received.

Once you’ve understood the basics of sensors, motors, and networking with Circuit Playgrounds, the possibilities for fun, educational projects are limitless. Good luck as you continue to explore the possibilities that these amazing tools have to offer!