

# Emulating a USB Keyboard with Arduino Leonardo

## Make a "Google Bike"

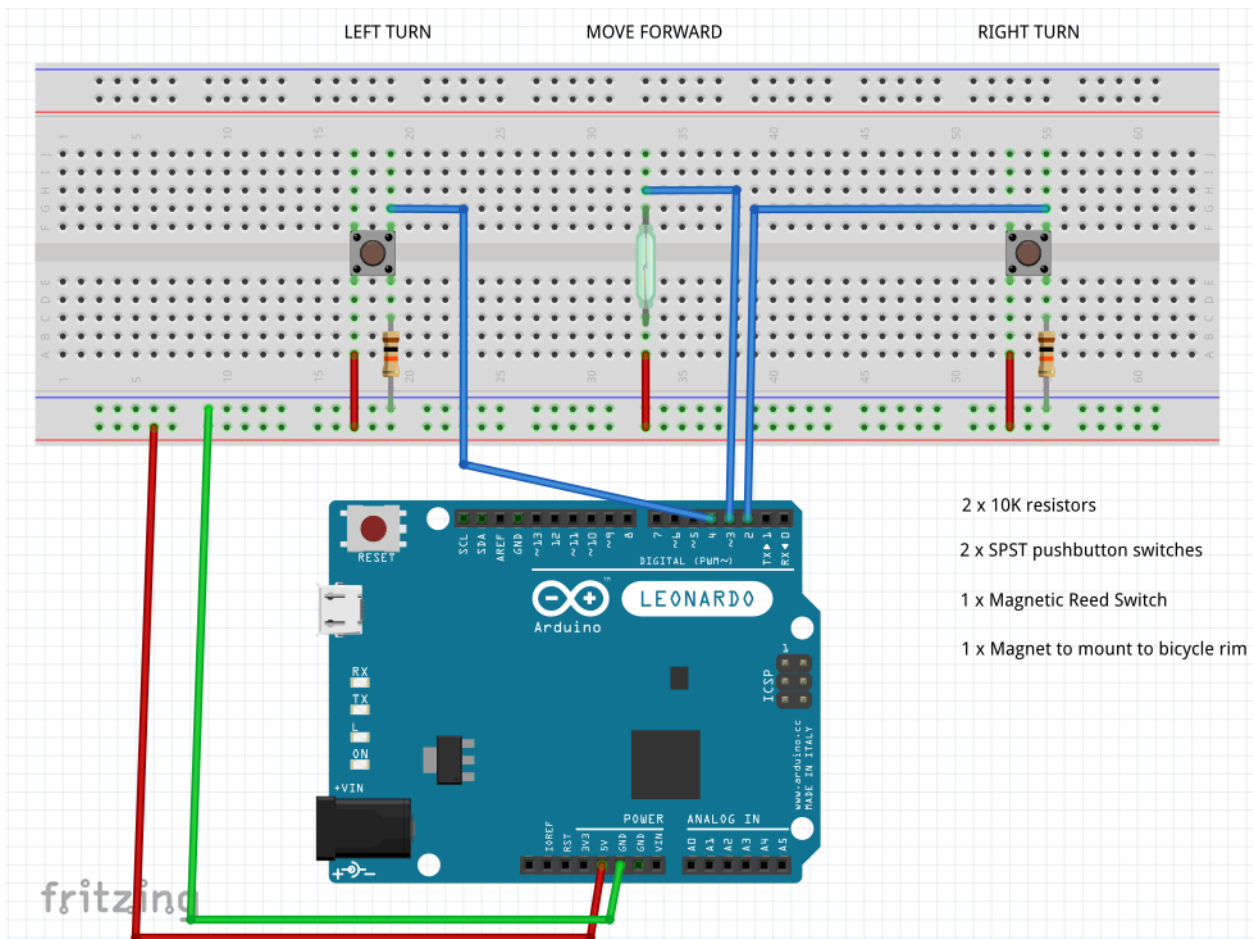
(G Payne – 2016)

The Arduino Leonardo is unique because it has a USB interface that allows you to send USB keyboard keystrokes and mouse movements to your computer.

The example here is to make a "Google Bike". It is an interface that allows you to pedal through roads on Google Street View. It uses two pushbuttons for turning left and right and a magnetic reed switch and magnet mounted on the wheel of a stationary bike for signaling Street View to move forward down a street.

### Parts Needed:

- Arduino Leonardo board and breadboard (or any Arduino with USB emulator capability)
- 2x Momentary Contact Switches or buttons
- 2x 10K resistor
- 1x Magnetic Reed Switch
- Connecting wires



For simplicity, the diagram is shown compactly on the breadboard. However in practice, you will want to mount the reed switch on the bicycle frame and have wires running back to the Arduino. Same goes for pushbuttons you want to mount on the handlebar of the bicycle.

When mounting the reed switch, you want to orient it so that having a magnetic pass by it will activate the switch. It's helpful to use a digital multimeter set to resistance, so you can watch the display to see when the resistance goes to zero. Rotate the barrel of the switch and keep trying the magnet it on it until you see the resistance go to zero. That will be the correct orientation of the switch relative to the path of the magnet.

To emulate a USB keyboard, you need the 'Keyboard.h' library included in your sketch. Should you wish to emulate a USB mouse, you'll need the 'Mouse.h' library included.

### **How does Google Street View work?**

If you go to Google Maps and select a street view at the bottom of the screen, you'll be given a street level view of the area. By using the UP arrow you can move forward through the scene. The LEFT and RIGHT arrows will turn your view left or right and will also allow you to turn left or right onto a new street. Alternatively, you could use 'a' for left and 'd' for right.

In this sketch, we will have the Leonardo send a KEY\_UP\_ARROW to the Google Street View every time we have two rotations of the bicycle wheel. Similarly, it will send KEY\_LEFT\_ARROW and KEY\_RIGHT\_ARROW to the Google Street View every time you want to turn left or right.

### **Why don't the KEY\_LEFT\_ARROW and KEY\_RIGHT\_ARROW keystrokes work?**

This was frustrating and took a while to figure out. It seems that Google Street View reacts immediately to the UP arrow key press, but the LEFT and RIGHT arrows (or a/d) take longer. So we put a delay into the sketch for the left/right commands and that seems to work well.

### **I'M STUCK!!! My Sketch is running away crazy! What do I do?**

One thing you must bear in mind is that when you are emulating a keyboard or mouse, you could inadvertently have your sketch run wild and start doing crazy things on your screen. You have to disconnect the Arduino from your computer to regain control.

But now you have to remove your problematic sketch from the Arduino.

Try this:

1. With the Arduino disconnected, open the Bare Bones Arduino sketch from Examples (or just an empty sketch).
2. Connect the Arduino back up to your computer and wait until you hear the chimes to indicate that it has been connected to your USB port.
3. Quickly HOLD DOWN the RESET key on the Arduino. Don't let go.
4. Click the UPLOAD button on your screen. Watch as Arduino compiles your code. As it gets close to the end, LET GO of the Reset button and it should upload the empty sketch to the Arduino and overwrite your incorrect sketch.

This may take a few tries to get your timing just right.

Enter the Arduino Sketch on the next page

```

// Controls for doing a virtual bike ride through Google StreetView
// Gordon Payne 2016

#include "Keyboard.h"    // library containing USB keyboard code

const int buttonRtPin = 2; // turn right
const int buttonUpPin = 3; // move forward
                          //(magnetic reed switch connected
                          // to stationary bicycle
const int buttonLftPin = 4; // turn left
long lastClkUp = millis(); // for measuring times between clicks
                          // - debouncing

long lastClkLeft = millis();
long lastClkRight = millis();
int btnStateUp; // for checking the state of a pushButton
int btnStateLeft;
int btnStateRight;
char upKey = KEY_UP_ARROW;
char leftKey = KEY_LEFT_ARROW;
char rightKey = KEY_RIGHT_ARROW;
int cyclePerMove = 2; // number of rotations before
                      // advancing forward - you may wish to adjust

int cyclCount = 0;
void setup() {
  // make the pushButton pin an input:
  pinMode(buttonRtPin, INPUT);
  pinMode(buttonLftPin, INPUT);
  pinMode(buttonUpPin, INPUT);
  // initialize control over the keyboard:
  Keyboard.begin();
  pinMode(13, OUTPUT);
  digitalWrite(13, LOW); // just turn off
                          // onboard LED - not needed
}

void loop() {
  // UP BUTTON
  // get the current state of the button pin
  btnStateUp = digitalRead(buttonUpPin);
  if (btnStateUp == HIGH) { // if button is pressed
    if (millis() - lastClkUp > 50) { // if it's been at least
                                      //50 milliseconds since last click
                                      // - debouncing

      cyclCount = cyclCount + 1;
      if (cyclCount == cyclePerMove) {
        Keyboard.press(upKey); // move forward in Street View
        cyclCount = 0;
        Keyboard.releaseAll();
      }
      lastClkUp = millis(); // reset lastClk
    }
  }
}

```

```

        //to the current time to be ready
        //for next click
    }
}
// RIGHT BUTTON
btnStateRight = digitalRead(buttonRtPin); // turn right
if (btnStateRight == HIGH) { // if button is pressed
    Keyboard.press(rightKey);
    delay(15);
    lastClkRight = millis();
    delay(400); // allow for Google Street View to
                // respond to right turn input
    Keyboard.releaseAll();
}
// LEFT BUTTON
btnStateLeft = digitalRead(buttonLftPin);
if (btnStateLeft == HIGH) { // turn left
    Keyboard.press(leftKey);
    delay(15);
    lastClkLeft = millis();
    delay(400);
    Keyboard.releaseAll();
}
delay(10); // slow down sketch for easier viewing on screen
}

```

Upload this sketch to Arduino and go to Google Maps and a Street View scene.

Once you have your hardware built, you should be able to pedal your way down a street and turn left or right onto other streets. Have fun!

For more fun, hook up an LCD panel to the Arduino and write code to show your speed and Watts of power being generated as you pedal. You could also display the kilometres or miles driven. You just need to do some calculations with Pi and the radius of your bicycle wheel to get distances and then convert to the units you want. You can convert the rotation counts and the time elapsed into the speed. Similarly, you can use simple formulas for power to get Watts of energy you are generating. Put a Reset button on your board if you want to be able to reset your trip odometer.

More information about Keyboard and Mouse control with Arduino Leonardo can be found at:

<https://www.arduino.cc/en/Reference/MouseKeyboard>

There are also excellent examples in your Arduino IDE under the File/Examples/USB menu

**Now go out and MAKE SOMETHING AMAZING!!!!**