

Light Navigator

*A simple Arduino project using a 5-position
switch sensor and an 8x8 LED display.*

Ryan Hardy
Marisa Smith
Nicholas Warren

Table of Contents

Introduction	2
The Microcontroller Platform	3
The Test Device	4
Development tools	5
The Experiment	6
Conclusions	7
Contributions	8
Project code	9
Sources	9

Introduction

(Author: Marisa Smith)

Using an 8x8 LED display (MAX 7219) and a 5-position switch sensor (basically a simple joystick) controlled by an Arduino, we plan to create a simple game. Ultimately, we would like to use the joystick to direct the player character, or pip, represented by a blinking LED around a maze displayed on the 8x8 board. The pip will be able to move in 4 directions. There are a few steps along the way:

1. get the 8x8 display working [done]
2. create a level [done]
3. be able to move the pip in the level [done]
4. use input from joystick [done]
5. failure states [done]
6. win screen
7. more levels

Depending on how well this project is scoped and how hard each step is, we might have to scale back.

The Microcontroller Platform

(Author: Ryan Hardy)

Explain the Arduino here

[Describe the microcontroller board you are using. This description should indicate where such boards can be obtained, and lay out the basic architecture of the board. You can paste screen captures of diagrams from any documentation you find online. You should describe the basic capabilities of the actual processor on the board. Show a block diagram of the device indicating any special features you find that are interesting and useful in conducting your experiment.]

The Test Device

(Author: Nicholas Warren)

MAX7219 Red Dot Matrix Module

27801 5-Position Switch

[wiring diagram]

[Each project uses some kind of device that ends up being controlled by the microcontroller. Describe that device and give an overview of what it might be used for. Basically, describe what is involved in interacting with the device. Again, use screen captures of diagrams from documentation is you can.]

Development tools

(Author: Nicholas Warren)

The LEDControl Library was very useful for setting up our LED Matrix. It sets up an LedControl object that has various functionality. We, however, only used 4 methods. Three of these methods were only used for setup and the majority of the work was done with setLed. This function allowed us to specify the exact LED we wanted to work with.

The LedControl object would assign the pins passed to be output and create an array of bytes used to set the rows of up to 8 separate devices. Each time an LED is set you are actually setting the whole row. It then, does a display test to see if all LEDs are functional, sends a decode message to see if the Matrix is functioning properly, clears all LEDs, and finally goes into shutdown.

The function setLed turned on or off the LED at a row by column. The parameters are addr, an int designating the device you wanted to use(always 0 for us), the row of the LED, the column of the LED and a boolean for on or off. So, setLed creates a byte in which zeros represent off LEDs and ones represent on LEDs. This byte is then sent to shiftOut.

turn called shiftOut, an arduino.h function, to send data directly to the LED Matrix. To call this function the CS pin needs to be set to low, telling the device to get ready to receive data. Then, call shiftOut with the parameters of dataPin (DIN), clkPin (CLK), bitOrder (most bit first) and the value(a byte). First you must send in the opcode, which is the number of the row you want to manipulate, then you send in the byte that sets the LEDs in the row. Once, you are done calling shiftOut, you must turn the clkPin to High so the device knows to stop reading.

The Experiment

(Author: Marisa Smith)

We were required to use at least one sensor in this project, but it seemed more fun and interesting to have two affect each other. So, our original idea was to use the 8x8 display and joystick combined. The obvious choice for these components was a simple game; we envisioned a lit dot moving around on the display, controlled by the user.

As we continued discussing, we thought the next step could be to make a maze for the player to navigate around to an "exit," which would be the win condition. We also discussed that touching a wall would reset the position of the dot, to add some challenge and a failure condition.

To set up the 8x8 LED display, we used these connections (Arduino - Display):

- 12 - DIN
- 10 - CS
- 11 - CLK
- 5V - DCC
- GND - GND

The 5-position switch sensor was set up as follows (Arduino - Switch):

- 9 - UP
- 8 - LT
- 7 - RT
- 6 - DN
- 3.3V - VCC
- GND - GND

Conclusions

(Author: Marisa Smith)

One of the challenges of this project was debugging. Because the code ran on an Arduino, there was no “easy” way to check what was in a variable through a print statement. The 8x8 board was easier to get working because a lot of people have played with something similar and written about it, and we could instantly see if the part was wired to the Arduino properly, because all of its lights lit up.

For the switch, in the beginning, we didn’t know if the switch was broken or just wired incorrectly. It was also a less popular choice for projects so there was no template we could look at as a guide that was known to work.

Other ideas that have stemmed from this project include:

- an Arduino board that fetches a weather report (somehow?) and displays a corresponding graphic on the 8x8 display
- adding teleports to the game
- finding a board that has multicolored LEDs and using the different colors to represent different things in the game
- maps that change over time
- adding sound with a third device

[What did you learn about hardware control from this project? Did you get any ideas for other projects you might try in the future?]

Contributions

Joystick Controls - Ryan & Nicholas

Project Manager - Marisa

Level Designer - Marisa

Documentation Formatting - Marisa

Game Code - Marisa & Nicholas

Wiring - Nicholas

Requirements - Roie Black

Project code

[Include a listing of the code you used for this project.]

Sources

<https://www.parallax.com/sites/default/files/downloads/27801-5-Position-Switch-v1.1.pdf>

<http://www.instructables.com/id/LED-Matrix-with-Arduino/>

<https://www.arduino.cc/reference/en/language/functions/advanced-io/shiftout/>

<https://www.arduino.cc/reference/en/language/functions/digital-io/pinmode/>

<https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>