20150330 POV Persistence of Vision Display

[Original Article](http://www.instructables.com/id/Persistence-of-Vision-Wand/?ALLSTEPS)

Picture 1.



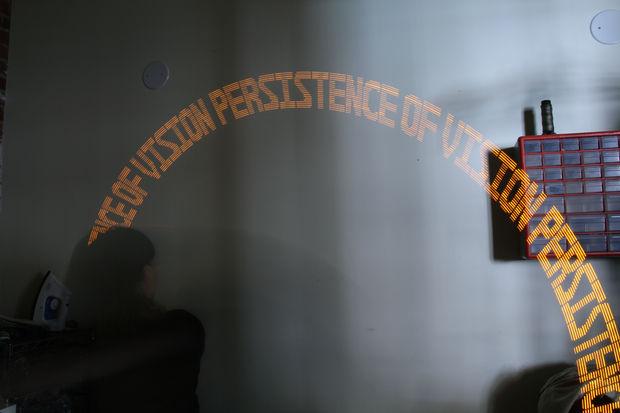
Picture 2



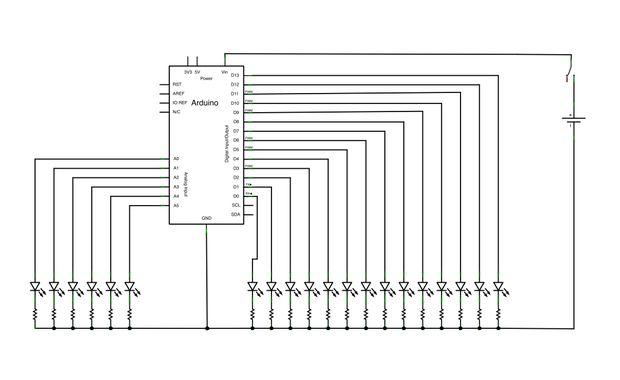
Picture 3.



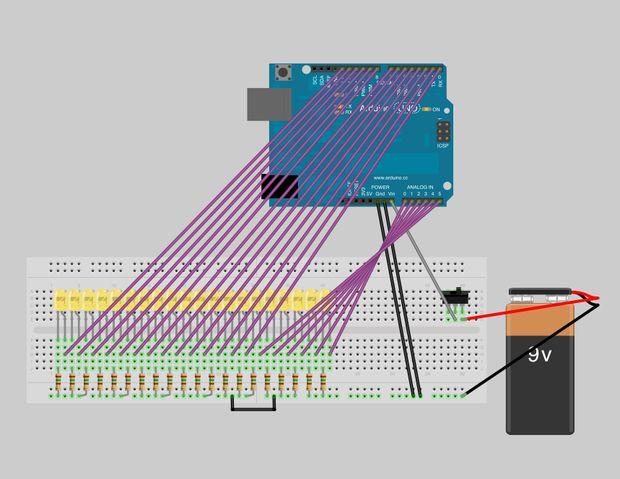






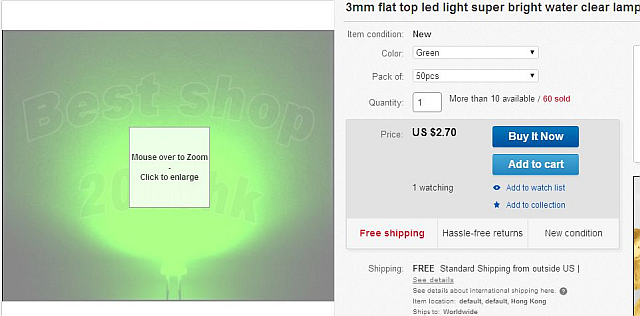
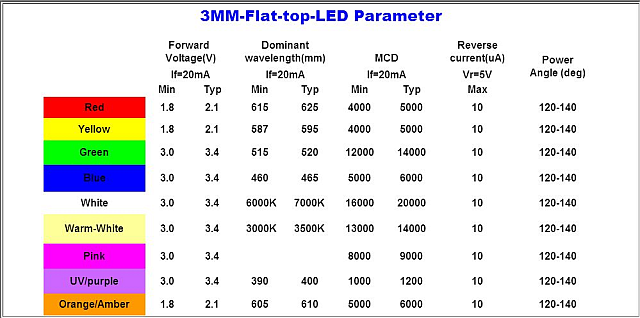
PiP

Picture 4.



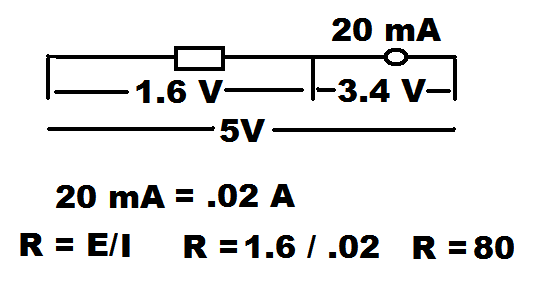
========================== PRESENTATION ==========================

**POV1** [**POV1**](https://drive.google.com/drive/#folders/0BySJuScrlj2OYWE0YTk3MDAtOWFlMi00NTMxLWI4NGEtZjY3ZDZhZWE0N2M2/0BySJuScrlj2OV1p3VHp3QjR3TzA)vvvvvv Persistence of Vision (P. O. V.) Wands are a fun way to create interesting long exposure photographs and light displays. The wand consists of a single row of L E Ds controlled by an Arduino Uno. When the wand is turned on it sequences through all the lights (does a test) and then they appear to flicker in a random pattern. If you move the board quickly back and forth, you will see text or an image appear. This wand project is incredibly customizable. Jims corner has all the documentation, and schematics. Jim has created a breadboard version for this demonstration. Jim purchased high viewing angle high brightness L E Ds from E bay. Feel free to choose your own color L E Ds, write personalized messages, and maybe even add something new to the project!  
^^^

vvv  
**POV2** Jims Configuration.  
Jim used 120 ohm resistors on his **green** LEDs so they wouldn't be operating above their maximum ratings. The exact LEDs are shown in figure 5. Figure 6 shows the ratings for various color LEDs. For the green LEDs note the Typical forward voltage of 3.4 Volts and the brightness rating at 20 milliamps inferring the normal operating current is 20 milliamps.  
 ***Picture 5*.** [Ebay LED order address](http://www.ebay.com/itm/121394446847?_trksid=p2060353.m2749.l2649&var=420333482742&ssPageName=STRK%3AMEBIDX%3AIT)  
 vvv ***Picture 6*** show the ratings for the LEDS Jim purchased from Ebay..  
^^^  
**POV3**  
Check the datasheet of your L E Dees to calculate your values. The L E Dee Operating voltage (shown above) was about 3.4 Volts and the operating current was 20 milliamps.

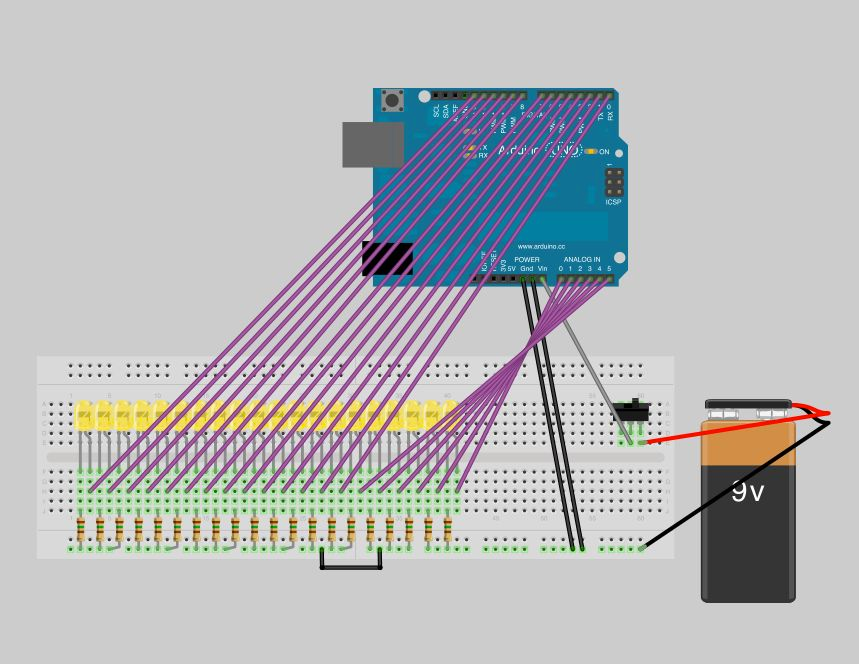
^^^

**POV3.5**  
vvv So we needed a voltage drop of about 1.6 volts (using a 5 volt supply) at 20 milliamps. So Resistance equal Voltage over Current or 1.6 volts divided by .02 leaving us with 80. I only had a couple of 82 ohm resistors so I used 120 ohm resistors instead (reducing the brightness somewhat). ( I immediately ordered more 82 ohm resistors for Leroy).



^^^

**POV 3.6**vvv  
I mounted everything on a breadboard similar to the drawing provided in the  
article (Figure 9) and used a 9 volt battery as the supply voltage. Figure 10 shows Jims version.

Figure 9.  
Figure 10.

^^^  
**POV 4.**  
vvv  
I mounted the display on a ceiling fan and fired it up (See Figures 7 and 8). Using the Grand Poo Bah’s amazing brainpower an ingenious method was used to attach the display to the fan blade. Grass trimmer line was used to attach the board to the inner fan blade arm (looping it around the arm once) and even more ingenious, a rubber band was used on the outer blade. The rubber band was handy as our genius did not install a switch on the front of the unit and the rubber band allowed temporary access to the battery to power it up or down. After adjusting the fan speed and the speed of the display a couple of times I was able to get some reasonable pictures. The default speed for Jims CheapO camera was a. s. a. 1600. Jim adjusted the film speed to a. s. a. 400 to get a longer exposure time, thus including all of the text in one shot.  
Pictures 7 and 8.

  
^^^

**POV5**  
vvv  
Picture 1. displays the word “AT CLUB” twice. You should note the certer of the picture is the hub of Jims ceiling fan.



**POV6**vvv  
Picture 2. displays the word “Jim”.

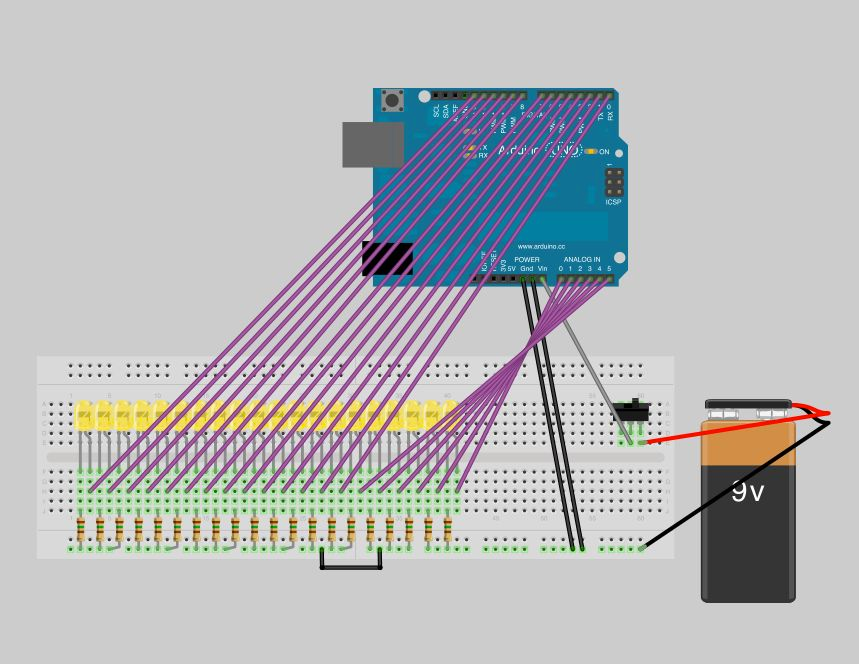


**POV7**  
Picture 3. displays the word “Gail”.



----------------------------------------------------------------------------------------------------------------

I built a breadboard similar to the one below (and loaded the code from the next section



Code I am working with:

//POV Wand Arduino code

//by Amanda Ghassaei

//instructables.com/amandaghassaei

//April 2012

//for use with arduino uno or duemilanove with atmel328

/\*

\* This program is free software; you can redistribute it and/or modify

\* it under the terms of the GNU General Public License as published by

\* the Free Software Foundation; either version 3 of the License, or

\* (at your option) any later version.

\*

\* This program is distributed in the hope that it will be useful,

\* but WITHOUT ANY WARRANTY; without even the implied warranty of

\* MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the

\* GNU General Public License for more details.

\*/

//in most of this code I have used the arduino portpin assignments to send data to pins, you can read more about how that works here: http://www.arduino.cc/en/Reference/PortManipulation

//I've also included (and commented) the standard arduino library commands that perform the same functions and the port commands in case you are interested

#include <avr/pgmspace.h>//need to store letter arrays in flash memory- or else we run out of space, more info here: http://arduino.cc/en/Reference/PROGMEM

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

THIS NEXT SECTION IS WHAT YOU'LL WANT TO EDIT TO CREATE YOUR OWN MESSAGES

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

// setup

String povtext = "POV";//PUT YOUR MESSAGE HERE!!- must be in all caps, spaces are fine, no punctuation

byte refreshrate = 10;//delay time for pixels to refresh in milliseconds- experiment with different values

//get length of string povtext

int dimtext = povtext.length();

//letterArray to make sure firmare is loaded correctly- each led should light up in order upon turning on

boolean load[]= {

1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,

0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,

};

//incoming data storage

byte data1 = 0;//for portB

byte data2 = 0;//for portC

byte data3 = 0;//for portD

//variables

byte n; //variable for loops

byte t; //variable for loops

byte l; //variable for loops

//The letters of the alphabet- edit the look of these if you want, just make sure the letters m and w are 15 pixels wide and the rest are 12 pixels

boolean letterA[] PROGMEM = {

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

};

boolean letterB[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

};

boolean letterC[] PROGMEM = {

0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1,

1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0,

};

boolean letterD[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,

};

boolean letterE[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

};

boolean letterF[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

};

boolean letterG[] PROGMEM = {

0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0,

};

boolean letterH[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

};

boolean letterI[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

};

boolean letterJ[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

1, 1, 1, 0, 1, 1, 1, 1, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,

0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,

0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,

};

boolean letterK[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

};

boolean letterL[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

};

boolean letterM[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

};

boolean letterN[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

};

boolean letterO[] PROGMEM = {

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

};

boolean letterP[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

};

boolean letterQ[] PROGMEM = {

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1,

};

boolean letterR[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 0,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

};

boolean letterS[] PROGMEM = {

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,

0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,

0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0,

0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0,

0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,

0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1,

0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

};

boolean letterT[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

};

boolean letterU[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

};

boolean letterV[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0,

0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,

0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

};

boolean letterW[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1,

0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0,

0, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0,

0, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0,

0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0,

};

boolean letterX[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0,

0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,

0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,

0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

};

boolean letterY[] PROGMEM = {

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

1, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 1,

0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0,

0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0,

0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0,

0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

};

boolean letterZ[] PROGMEM = {

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0,

0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0,

0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0,

0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 0, 0,

0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0,

0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0,

0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 0, 0,

0, 0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0,

0, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0,

0, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,

};

void sendToWand(boolean letterArray[]){//function to get array data

for (t=0; t<12; t++){ //for each time step

for (l=0; l<6; l++){ //for first six rows of data

data1 = data1 << 1;//bitwise shift left

data1 |= pgm\_read\_byte\_near(letterArray + (l\*12+t));//add next value from dataset

}

for (l=6; l<14; l++){ //for next eight rows of data

data2 = data2 << 1;//bitwise shift left

data2 |= pgm\_read\_byte\_near(letterArray + (l\*12+t));//add next value from dataset

}

for (l=14; l<20; l++){ //for next six rows of data

data3 = data3 << 1;//bitwise shift left

data3 |= pgm\_read\_byte\_near(letterArray + (l\*12+t));//add next value from dataset

}

//SET PINS:

PORTB = data1;

PORTD = data2;

PORTC = data3;

delay(refreshrate);

//clear data storage

data1=0;

data2=0;

data3=0;

}

}

void sendToWandMW(boolean letterArray[]){//M and W are extra wide- they have a special function to get array data (with 15 time steps instead of 12)

//send data to leds via port/pin manipulation

for (t=0; t<15; t++){ //for each time step

for (l=0; l<6; l++){ //for first six rows of data

data1 = data1 << 1;//bitwise shift left

data1 |= pgm\_read\_byte\_near(letterArray + (l\*15+t));//add next value from dataset

}

for (l=6; l<14; l++){ //for next eight rows of data

data2 = data2 << 1;//bitwise shift left

data2 |= pgm\_read\_byte\_near(letterArray + (l\*15+t));//add next value from dataset

}

for (l=14; l<20; l++){ //for next six rows of data

data3 = data3 << 1;//bitwise shift left

data3 |= pgm\_read\_byte\_near(letterArray + (l\*15+t));//add next value from dataset

}

//SET PINS:

PORTB = data1;

PORTD = data2;

PORTC = data3;

delay(refreshrate);

//clear data storage

data1=0;

data2=0;

data3=0;

}

}

// //SAME AS ABOVE BUT WRITTEN USING ARDUINO LIBRARY COMMANDS:

//

//// for (t=0; t<12; t++){ //for each time step of matrix

////

//// for (int pin= 13; pin>=0; pin--) {//this loop sets digital pins 0-13 either high or low depending on the value of the corresponding matrix element

//// if (letterArray[(13-pin)\*12+t]) {//if matrix element == 1 then turn led on

//// digitalWrite(pin, HIGH);

//// }

//// else {//if matrix element == 0 then turn led off

//// digitalWrite(pin, LOW);

//// }

//// }

////

//// //the following lines address the analog pins indivially and set them either high or low depending on matrix element

//// if (letterArray[14\*12+t]) {//if matrix element == 1 then turn led on

//// digitalWrite(A5, HIGH);

//// }

//// else {//if matrix element == 0 then turn led off

//// digitalWrite(A5, LOW);

//// }

//// if (letterArray[15\*12+t]) {

//// digitalWrite(A4, HIGH);

//// }

//// else {

//// digitalWrite(A4, LOW);

//// }

//// if (letterArray[16\*12+t]) {

//// digitalWrite(A3, HIGH);

//// }

//// else {

//// digitalWrite(A3, LOW);

//// }

//// if (letterArray[17\*12+t]) {

//// digitalWrite(A2, HIGH);

//// }

//// else {

//// digitalWrite(A2, LOW);

//// }

//// if (letterArray[18\*12+t]) {

//// digitalWrite(A1, HIGH);

//// }

//// else {

//// digitalWrite(A1, LOW);

//// }

//// if (letterArray[19\*12+t]) {

//// digitalWrite(A0, HIGH);

//// }

//// else {

//// digitalWrite(A0, LOW);

//// }

////

////

//// delay(refreshrate);//delay after each column of data is displayed

//// }//go to next time step

void setup() {

//port/pin assignments- set all pins to output- more info here: http://www.arduino.cc/en/Reference/PortManipulation

DDRB = 0xFF;//port b- digital pins 8-13

DDRC = 0xFF;//port c- anlog pins 0-5

DDRD = 0xFF;//port d- digital pins 0-7

//the three lines above are the same as setting all pins as outputs using arduino library:

// pinMode(A0, OUTPUT);

// pinMode(A1, OUTPUT);

// pinMode(A2, OUTPUT);

// pinMode(A3, OUTPUT);

// pinMode(A4, OUTPUT);

// pinMode(A5, OUTPUT);

//

// for (int pin=0; pin<14; pin++){

// pinMode(pin, OUTPUT);

// }

//run intialization so we know device is working- leds should light up in order from top of wand to bottom

for (byte j=0; j<20; j++){ //for each time step

for (byte i=0; i<6; i++){ //for first six rows of data

data1 = data1 << 1;//bitwise shift left

data1 |= load[(i\*20+j)];//add next value from dataset

}

for (byte i=6; i<14; i++){ //for next eight rows of data

data2 = data2 << 1;//bitwise shift left

data2 |= load[(i\*20+j)];//add next value from dataset

}

for (byte i=14; i<20; i++){ //for next six rows of data

data3 = data3 << 1;//bitwise shift left

data3 |= load[(i\*20+j)];//add next value from dataset

}

PORTB = data1;

PORTD = data2;

PORTC = data3;

delay(100);

}

////SAME AS ABOVE LOOP, BUT USING ARDUINO LIBRARY:

// //turn on each LED one by one using arduino library commands

// for (int pin= 13; pin>=0; pin--){//turn on each digital pin sequentially for 100ms

// digitalWrite(pin, HIGH);

// delay(100);

// digitalWrite(pin,LOW);

// }

// //the following lines turn on each analog pin individually

// digitalWrite(A5, HIGH);

// delay(100);

// digitalWrite(A5,LOW);

// digitalWrite(A4, HIGH);

// delay(100);

// digitalWrite(A4,LOW);

// digitalWrite(A3, HIGH);

// delay(100);

// digitalWrite(A3,LOW);

// digitalWrite(A2, HIGH);

// delay(100);

// digitalWrite(A2,LOW);

// digitalWrite(A1, HIGH);

// delay(100);

// digitalWrite(A1,LOW);

// digitalWrite(A0, HIGH);

// delay(100);

// digitalWrite(A0,LOW);

//clear data storage

data1 = 0;

data2 = 0;

data3 = 0;

//clear ports- set all arduino pins to 0Volts

PORTB = data1;

PORTD = data2;

PORTC = data3;

}

void loop() {

//space at beginning of text

PORTB = 0;

PORTD = 0;

PORTC = 0;

delay(refreshrate\*3);

for (n=0; n<dimtext; n++) {//go through each character of povtext and call function sendToWand to display letter

if (povtext.charAt(n)=='A') {

sendToWand(letterA);

}

else if (povtext.charAt(n)=='B') {

sendToWand(letterB);

}

else if (povtext.charAt(n)=='C') {

sendToWand(letterC);

}

else if (povtext.charAt(n)=='D') {

sendToWand(letterD);

}

else if (povtext.charAt(n)=='E') {

sendToWand(letterE);

}

else if (povtext.charAt(n)=='F') {

sendToWand(letterF);

}

else if (povtext.charAt(n)=='G') {

sendToWand(letterG);

}

else if (povtext.charAt(n)=='H') {

sendToWand(letterH);

}

else if (povtext.charAt(n)=='I') {

sendToWand(letterI);

}

else if (povtext.charAt(n)=='J') {

sendToWand(letterJ);

}

else if (povtext.charAt(n)=='K') {

sendToWand(letterK);

}

else if (povtext.charAt(n)=='L') {

sendToWand(letterL);

}

else if (povtext.charAt(n)=='M') {

sendToWandMW(letterM);

}

else if (povtext.charAt(n)=='N') {

sendToWand(letterN);

}

else if (povtext.charAt(n)=='O') {

sendToWand(letterO);

}

else if (povtext.charAt(n)=='P') {

sendToWand(letterP);

}

else if (povtext.charAt(n)=='Q') {

sendToWand(letterQ);

}

else if (povtext.charAt(n)=='R') {

sendToWand(letterR);

}

else if (povtext.charAt(n)=='S') {

sendToWand(letterS);

}

else if (povtext.charAt(n)=='T') {

sendToWand(letterT);

}

else if (povtext.charAt(n)=='U') {

sendToWand(letterU);

}

else if (povtext.charAt(n)=='V') {

sendToWand(letterV);

}

else if (povtext.charAt(n)=='W') {

sendToWandMW(letterW);

}

else if (povtext.charAt(n)=='X') {

sendToWand(letterX);

}

else if (povtext.charAt(n)=='Y') {

sendToWand(letterY);

}

else if (povtext.charAt(n)=='Z') {

sendToWand(letterZ);

}

else if (povtext.charAt(n)==' ') {

PORTB = 0;

PORTD = 0;

PORTC = 0;

delay(refreshrate\*3);//off for 3 pixels

}

//space between each character

PORTB = 0;

PORTD = 0;

PORTC = 0;

delay(refreshrate);

}

//space at end of text

PORTB = 0;

PORTD = 0;

PORTC = 0;

delay(refreshrate\*3);

}