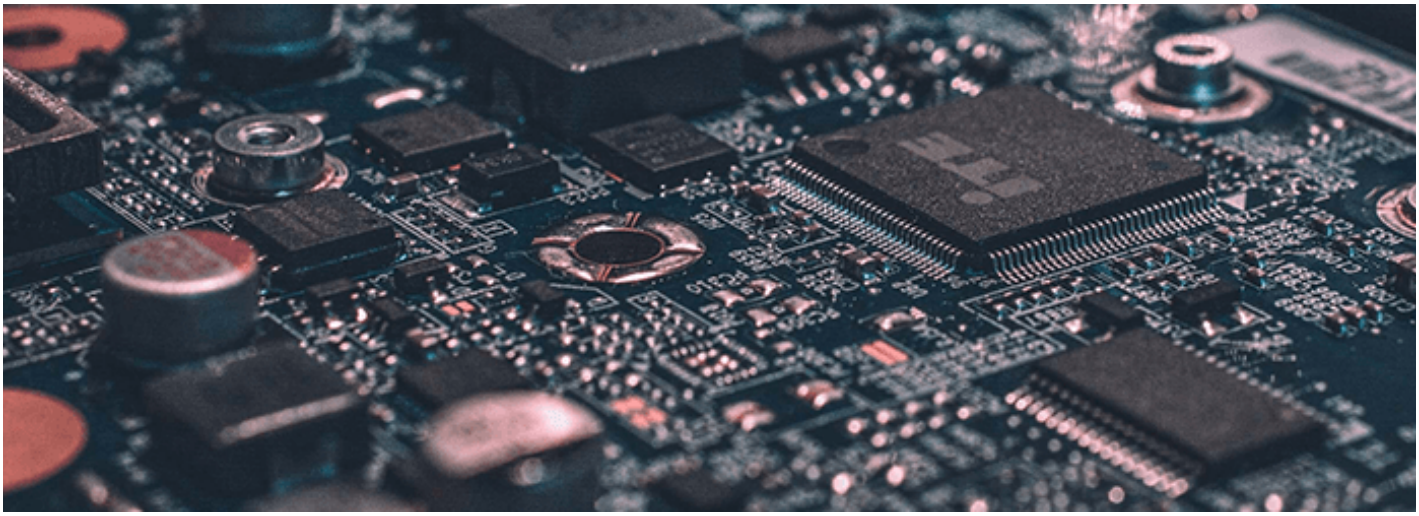
ACTIVE MATRIX. PASSIVE MATRIX. DIRECT DRIVE

How To Drive Displays



There are many different display technologies such as LCD, OLED, EPD, and ECD. They ar fundamentally different technologies with various driving requirements. That being sai some basic ideas of how to drive them. In this article, I will explain some fundamentals This information is relevant whether you are a professional engineer designing display hacker exploring seven segment displays for your Arduino projects, or simply if you are this topic.

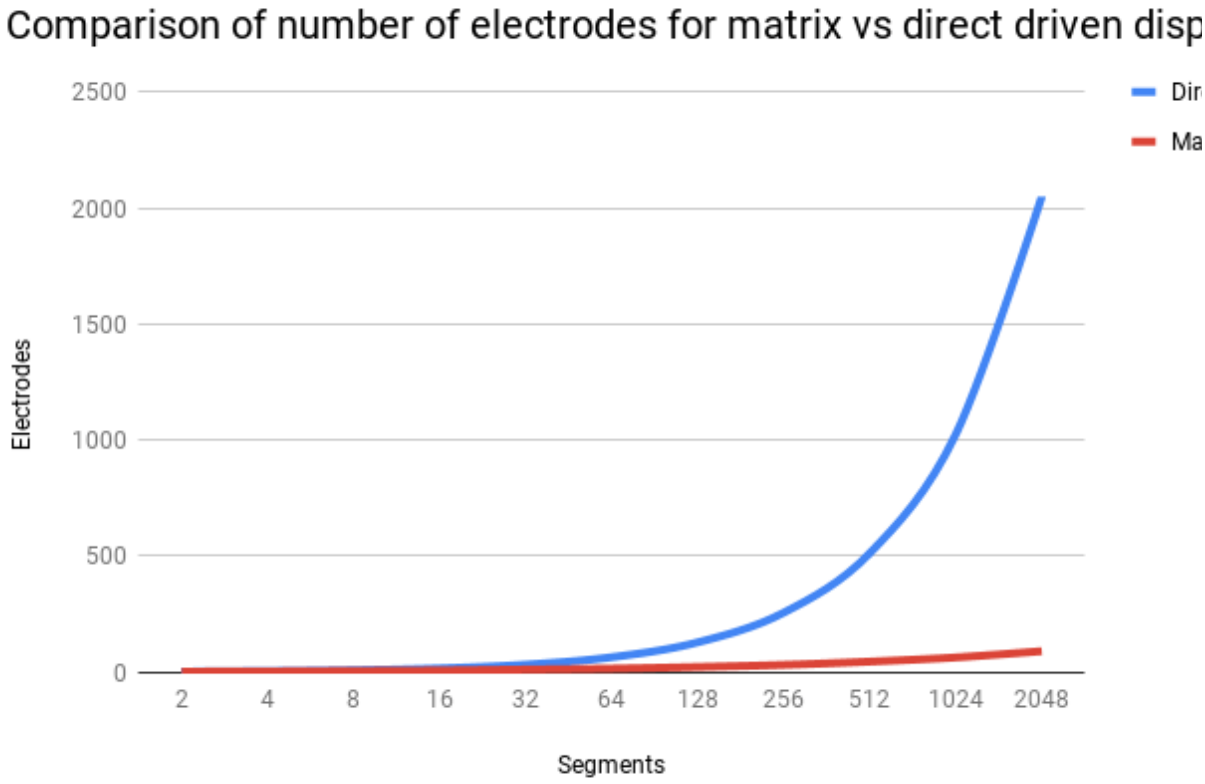
There are two main categories of displays; *segmented* (left picture) or *graphical* (right picture). In a segmented the display has to be predetermined (it can be numbers, **seven segment displays**, symbols, characters), while generic and can represent any given picture. The segmented displays are generally cheaper (even if they are cu graphical displays offer more flexibility in terms of the content displayed.

These display types can be driven by multiple methods with different degrees of complexity. A general guideli come with a higher driving complexity. There are two main types of driving, *direct drive* suitable for a low pixel that can handle millions of pixels.

Direct Drive

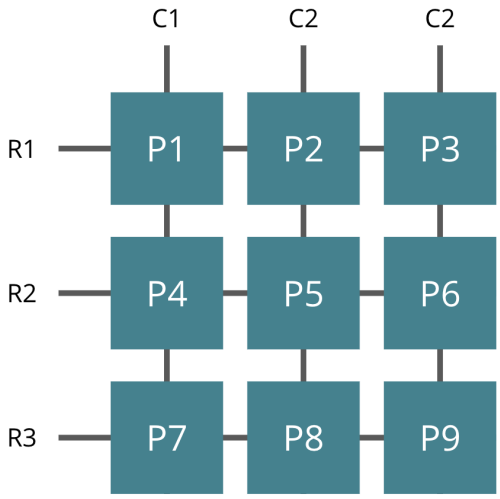
Direct drive is a common driving method for segmented displays such as seven segment displays. It is a very s each display segment is connected to a pin. A segment is addressed simply by setting a voltage to the targete cases, this type of display can be driven directly from many microcontrollers, eliminating the need for a dedic in turn, reduces the cost of the overall system. This is the case for the **Rdot display** which can be connected to with accurate driving voltage. An advantage with direct drive is the possibility to address all the pixels at the s direct drive is the driving method that offers the highest contrast.

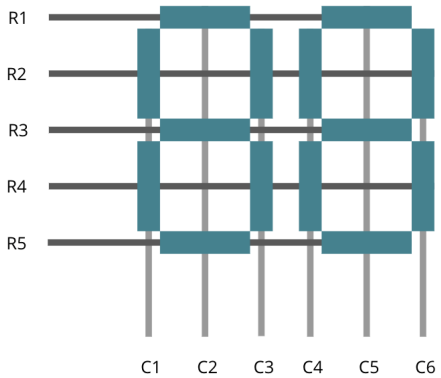
The drawback with direct drive is the high pin count for a display with lots of segments. For a display with mor direct drive is not recommended. From 30 to 100 segments, direct drive could be a good option. Below 30 segm very often the best choice. In the figure below I'm comparing direct drive and matrix drive in terms of electrod segment count.



Matrix Drive

To solve the issue with an unreasonable amount of electrodes, the segments, or pixels, has to be arranged in a
the figure below. This can be done with both segmented and graphical displays. For example, addressing pixel
addressing column 2 (C2) and row 1 (R1).





Passive Matrix

In a *passive matrix* display, pixels are addressed row by row, this is called time multiplexing. That means that a
updated first, then all pixels on row 2, etc.. meaning that for a display with three rows, each row is only addres

On retro displays, it is sometimes possible to see this effect as a continuous sweeping across the screen. For L contrast of the display which, in turn, limits the total number of rows possible. This method is often called *mu* segmented displays. Passive matrix drive is a cost-effective method to drive displays as it doesn't require any a However, just a few display technologies have the characteristics required for passive matrix drive.

The Rdot display technology has great potential to become the best passive matrix technology. This is because bistable (can maintain its state even when it is not addressed). This means that when one row is updated the update the other rows without losing the contrast on the first row. Passive matrix Rdot display is still under de planned to reach market during 2019. This display will be revolutionary to the display industry as it will be the display in high resolution and the first fully printed passive matrix display. In addition, it will still keep the low maintain the bendability. If you want to follow the development, sign up for our newsletter!

Any Questions? Just leave your email.

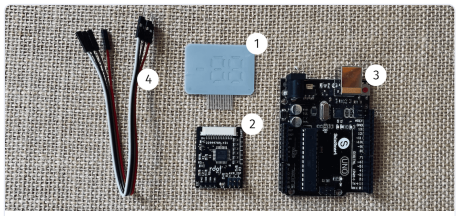
Please leave your email

Active Matrix

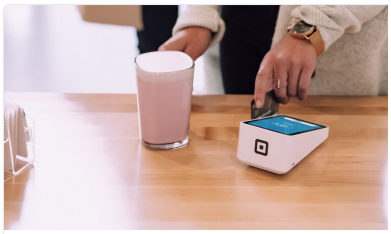
The sophisticated display, that you most likely are looking at right now, is based on active matrix technology. I

```
#include <RDOT_ECD_I2C_1.0.h>

int i2c_address = 43;         //The i2c address of driver board 4.1
int number_of_segments = 15;  //Number of segments on display (1-15)

RDOT_ECD ECD(i2c_address, number_of_segments); //ECD Object

void setup() {
}

void loop() {
  for(int i=-99; i<100; i++){ //Counting from -99 to 99.
    ECD.setNumber2x7(i);      //Set display to number i
    delay(5000);              //5 seconds delay between each update
  }
}
```

Driver 4 Library Documentation



Driver 4 Tutorial



RFID & NFC Electrochrom Display