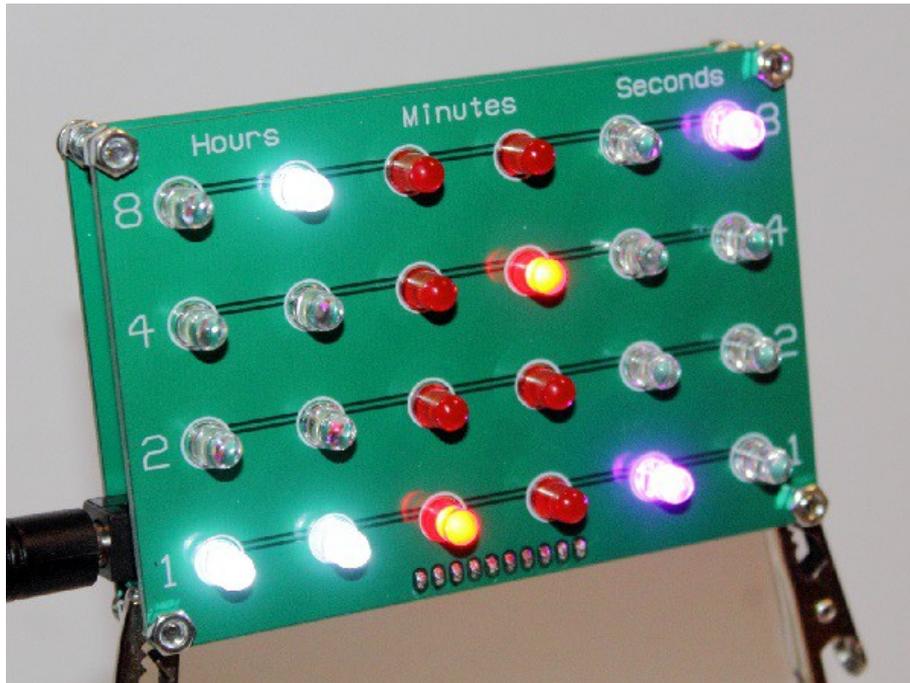


Raspberry-Pi Shield: Binary-Coded-Decimal Clock



SOFTWARE INSTALLATION INSTRUCTIONS

Driving the clock shield requires a small C program, running on a Linux distribution. I recommend a “Lite” Raspbian installation, as there's no point using a graphical OS when you're never going to see the display!

Operating system install

Download the latest “Lite” Raspbian distribution from here:

<https://www.raspberrypi.org/downloads/raspbian/>

(At time of writing, the latest version is Raspbian Stretch 2018-03-13.)

Write this to a memory card as usual (you'll need one that's at least 4GB in size) and boot your Raspberry-Pi with it.

As always, you login with a username of “pi” and a password of “raspberrypi”.

Network Setup

Wired LAN: If you're using a wired LAN connection, then the Pi will get all its connection details from your home/office router.

Wifi: If you're using a Wifi connection, you'll need to enter your network name and password. Run `sudo raspi-config` then select "Network Options" then enter your Wifi network name and password.

If you've entered it right, then running `ifconfig` will show that wlan0 has been configured, and has an IP address.

Activating SSH (optional)

If you think you'd like to access or configure your clock over your LAN, then it's worth activating the SSH daemon. This is optional. It's useful if you want to copy-and-paste the install commands (see the rest of this document) from your PC. If you don't want to do this, then it just means you'll need to use a screen and keyboard for the rest of the installation.

To activate the SSH server, run `sudo raspi-config`, select "Interfacing Options", then "SSH". Select "Yes" to enable it.

After this, you can connect to your clock from your PC with your usual SSH client.

Setting the time and timezone

When your Pi has a connection to the Internet, it will automatically set its internal clock to the right time and date. (Run `timedatectl` if you want to see a summary of this.)

If you run `date`, it'll tell you what it *thinks* the current time is. This will mostly likely be in UTC, as the Pi won't know what timezone you're in. To configure your timezone, run `sudo raspi-config`, select "Localisation Options", then "Change Timezone".

After configuring your timezone, run `date` again just to check your local time is now correct.

Installing the software

The only dependency this project has is Mike McCauley's rather excellent `bcm2835` library, which makes accessing the GPIO pins dead easy. Visit

<http://www.airspayce.com/mikem/bcm2835/>

for latest info.

To install it, just enter these commands, pressing return after each line:

```
cd
wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.55.tar.gz
tar xzf bcm2835-1.55.tar.gz
cd bcm2835-1.55/
./configure
make
sudo make check
sudo make install
```

Now we can install the BCD-clock software. Visit

<https://github.com/oddblk/bcd-clock>

for any up-to-date info.

Enter these commands to make a folder and download the source code into it:

```
cd
mkdir bcd-clock
cd bcd-clock
wget https://github.com/oddblk/bcd-clock/raw/master/bcd-clock.c
```

If you're using a first-edition Raspberry Pi then you'll need to modify the code, because the pin arrangements for GPIO was changed after the first commercial model. It's easy to do: open `bcd-clock.c` with a text editor (like `nano`) and find this line near the top:

```
static const int kPin_Max7219_CS = 27; // GPIO 27 (pin 13)
```

Change the 27 to a 21, then save it again.

Compiling the code

Enter this:

```
gcc -o bcd-clock bcd-clock.c -l bcm2835
```

This will make an executable called "bcd-clock" in the same folder.

Running the code

To test it, just run it like this:

```
./bcd-clock
```

Press ctrl+C to kill it.

Making the clock start on boot

To make it run on startup, we add it to the startup script:

```
sudo nano /etc/rc.local
```

Scroll down to the bottom of the file, and add this line just before the exit 0:

```
/home/pi/bcd-clock/bcd-clock &
```

Then save it. To prove it works, just reboot:

```
sudo reboot
```

And after rebooting, your clock should start automatically!

http://danceswithferrets.org/geekblog/?page_id=1034