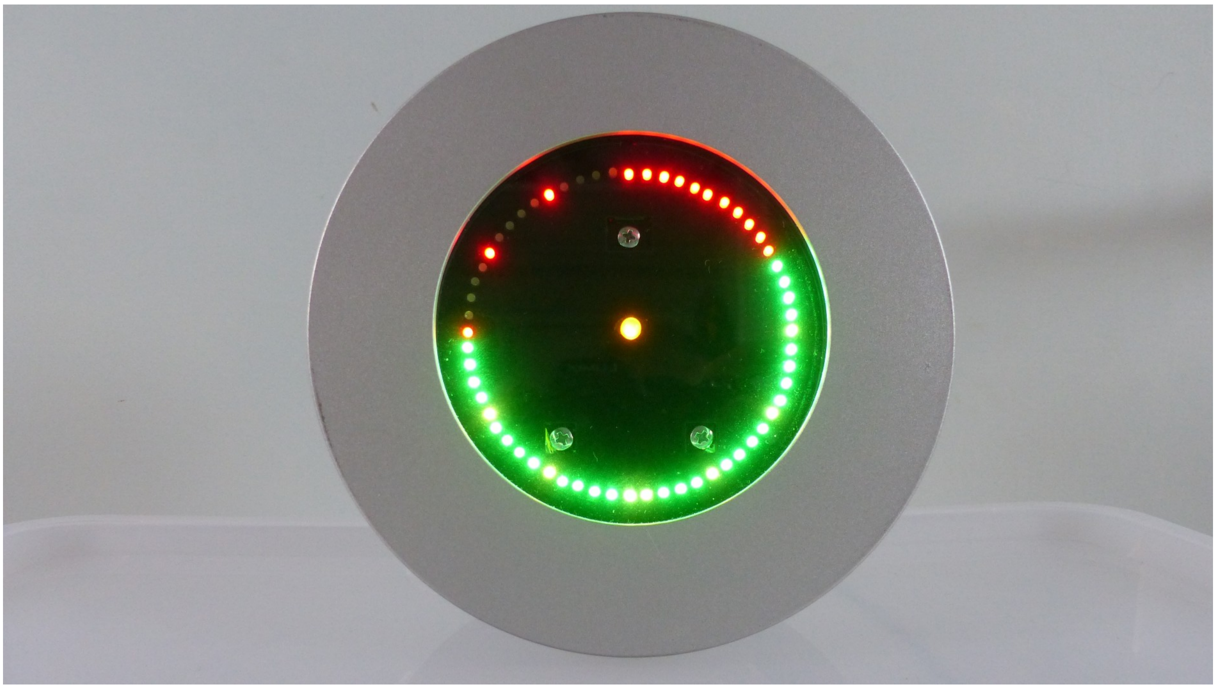
Analog Digital Clock



The reason for making this clock was because my original IKEA clock did not work anymore and I really liked the housing of this clock. I found it a waste to throw the clock away and decided to re-use it for an analog / digital clock.

I could have made a standard clock but I decided to make something different. Like any other clock it shows the time but not in a standard way. Using 60 bi-color red/green LEDs the clock shows the time. The red LEDs are used to show the hours and the green LEDs show the minutes. The seconds are indicated by a walking yellow (red + green) LED and by a blinking yellow LED in the center of the clock.

It requires some practice to know how to read the clock. Since the LEDs are used to show both the hours and the minutes it needs a special way of presenting the time. The time is shown as a bar of LEDs where the longest bar shows either the hours or the minutes. If the longest bar is presented by the hours then the shorter bar presents the minutes in green and the remaining part shows the hours in red. In order to make the clock more readable in cases the bars are short, I added an hour indicator using the red LED. If the minutes become larger than the hours, the bars swap, that is all previous green minutes become red to show the hours and the remaining part will show the minutes so in fact almost all green becomes red and the other way around.

It is somewhat difficult to explain how it works so please watch the video. Due to the multiplexing of the LEDs it looks as if the LEDs are blinking in the video. This is only captured by the camera, not by the human eye.

Video 1: https://youtu.be/gEIFPwg_oro

As always I built this project around my favorite micro controller the PIC, using the JAL programming language.

The designs

In total I made three different versions of the clock before I was satisfied. These versions were designed as follows:

1. Using a standard 20 MHz crystal for the PIC. With this design the clock was out of sync 1 second after one day of operation. This was too much. Next to that the time was lost when you switched off the clock since there was no back-up battery in the design.
2. Using a DS1302 clock module. The nice thing about this module is that it has a back-up battery so the time is not lost when you switch off the clock. When I tested the clock with this module the clock was out of sync 7 seconds! after one day. I think this is caused by either the wrong crystal or a bad PCB design.
3. Using a DS3231 clock module. This module also has a backup battery and it is more accurate than the DS1302. The clock worked fine with this module so I used this for the final design. Because of that, the PIC did no longer need a crystal.

The complete design is drawn-up in two schematic diagrams:

1. Clock Controller using the PIC and the LED driver using shift registers
2. 60 Bi-color LEDs, Red & Green

Required components

You need to have the following components for this project:

- A piece of breadboard
- PIC Microcontroller 16F1823
- 3 shift register 74HC595
- 1 Darlington Transistor Array ULN2803A
- IC sockets: 1 * 14-pin, 3 * 16-pin, 1 * 18-pin
- Clock module DS3231
- 2 push-button switches
- Resistors: 2 * 33k, 8 * 100 Ohm, 8 * 47 Ohm
- 1 electrolytic capacitor 100 uF/16V
- Ceramic capacitors 1 * 470 nF, 5 * 100 nF
- LEDs: 60 2 mm bi-color (red/green), 1 5 mm yellow
- Jack plug 3 mm
- LM7805 Voltage Regulator
- Optional: Headers for connecting the external parts to the breadboard
- A housing for your clock.

See the schematic diagrams on the next page on how to connect the components. It requires quite some soldering, especially for connecting the 60 LEDs.

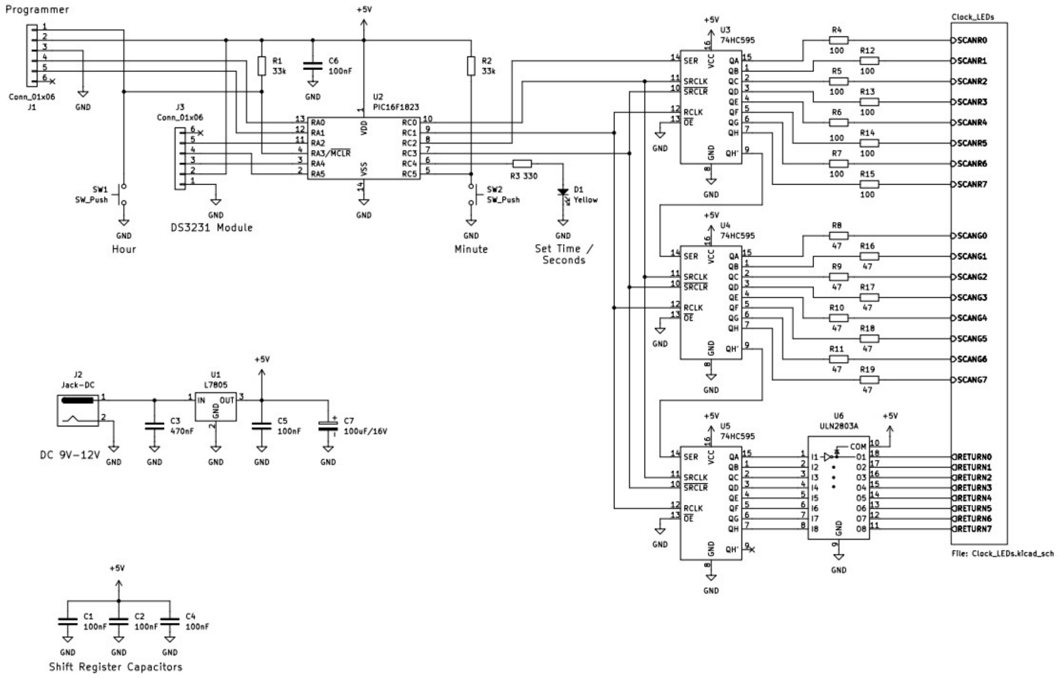


Figure 1: Controller and LED driver

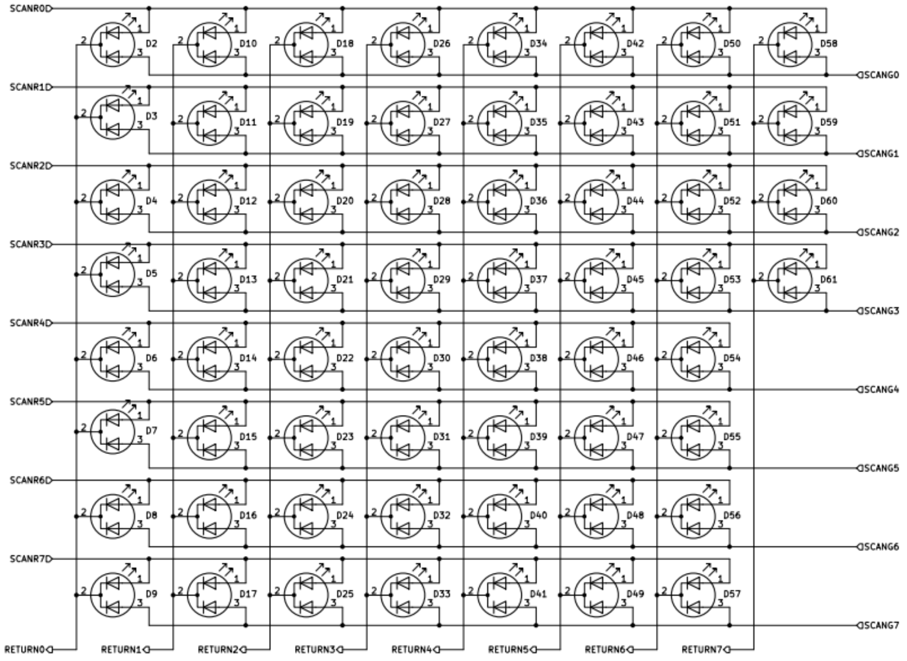
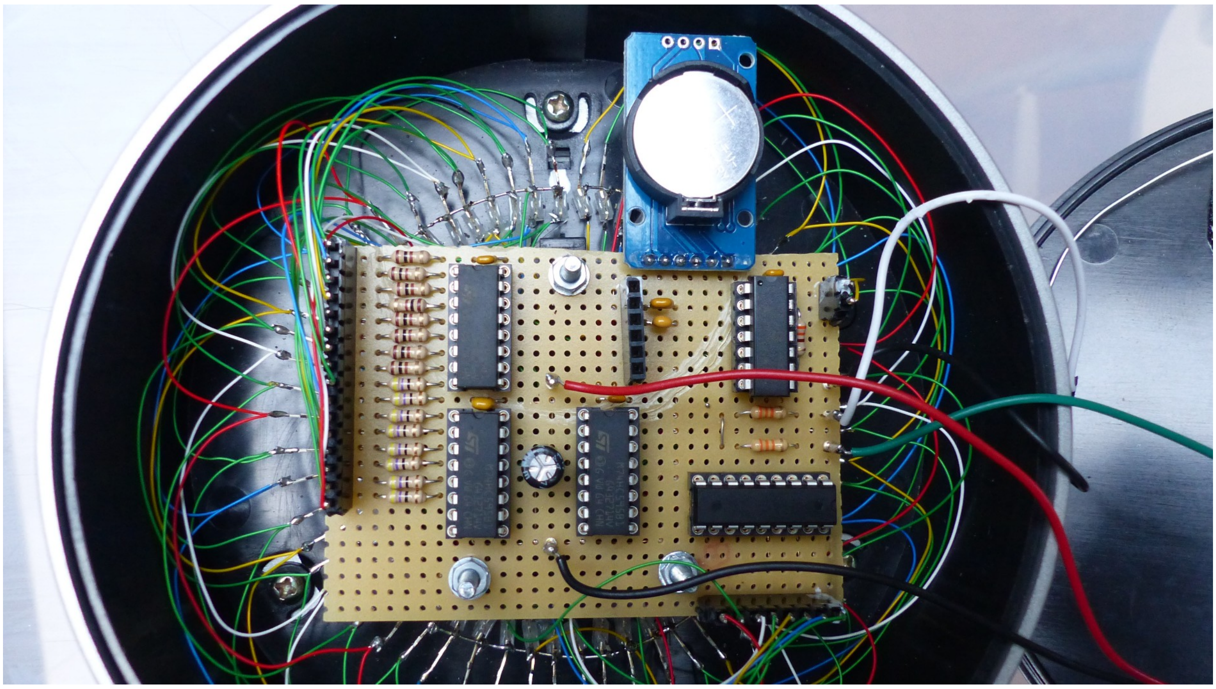


Figure 2: LEDs

Building the clock.

Have a look at the picture on how I finally build the clock. Note that the voltage regulator of 5 Volt is not in this pictures since it was added later.



I started by removing the internals of the original clock after which I drilled 60 holes of 2 mm for the bi-color LEDs in the front plate. Then I painted the front plate black and added a piece of plastic to cover the hole where the original hands of the clock where positioned. Now a yellow LED is located at that position.

Then I mounted all 60 LEDs, used some hot glue to keep them at their location and connected them with Kynar wire to each other. Last but not least I assembled the breadboard with all components.

On the back cover I mounted the two push-buttons and the power Jack. Forget about the extra plate I glued on the back as show in the picture. I added that because in my first design the push-buttons were located there but I needed to move them because I had to add the DS3231 module and I could only find a spot where those buttons were when I made my first design.

Initially the clock operated at 5 Volts but since I only had a 12 Volt power supply, I added an LM7805 later. I also added a header for connecting a PIC programmer but that connectors is optional.

The Software

As already mentioned, the software is written for a PIC16F1823 using the JAL programming language. The PIC runs on an internal clock of 32 MHz. As mentioned earlier, the clock timing is done by the DS3231 clock module.

The software performs the following main tasks:

- Initializing the DS3231 module using an I2C interface. The module will generate a 1 second signal which is connected to the interrupt pin of the PIC. The PIC uses this 1 second interrupt to read the time from the DS3231 module.
- Driving the 60 bi-color LEDs via the shift registers. In the schematic diagram it can be seen that the LEDs are connected in a 16 by 8 matrix. This reduces the number of wires that are needed to connect all LEDs. This matrix design requires that the PIC needs to multiplex the LEDs as to be able to light them up individually. Multiplexing the LEDs is done on an interrupt basis where the refresh frequency is 70 Hz so invisible to the human eye.
- Handling the push-buttons. These are used to set the time, one for setting the hours and one for setting the minutes. Both buttons need to be pressed shortly at the same time to activate the time-setting mode. When the time setting mode is selected the yellow led will be continuously on. After 5 seconds of not using the push-buttons the clock returns to the normal time operation and the yellow LED will start to blink.

See the second video on how to set the time.

Video 2: https://youtu.be/iKnxG3Wfvv8