**A circuit board with wires

Description automatically generated with low confidence**

**Counting Binary Lights by Harvey Buhagiar**

Over the Winter break, I’ve been trying out some of the coding exercises that the EbI team sent around and I started experimenting with some of the components in our lab kits, especially the Raspberry Pi Pico.

I thought I’d share with you one of the projects I came up with – the Counting Binary Lights Machine™!

This is a program which I designed to run from your Raspberry Pi Pico using its native language, MicroPython. The source code and breakdown thereof are provided and all you’ll need to get this working is a breadboard set up as I’ll instruct below and a computer to load the code onto your Pico.

**What it does**

Displays the numbers 0-15 using 4 binary digits represented by red LEDs. It has two programs, one which counts sequentially at a speed customisable by the user, *auto\_sequence(interval)* and the other which increments the counter by one at the press of a button on the breadboard, *button\_sequence()*.

**What you’ll need**

Everything you’ll need is provided in your Faculty of Engineering lab kits with the exception of a micro-USB cable which you’ll need to connect your Raspberry Pi Pico to your computer. You’ll need:

* 1 \* breadboard
* 12 \* jumper cables with stripped ends (cut from the multi-core cable)
* 1 \* button
* 1 \* Raspberry Pi Pico
* 4 \* 220Ω resistor
* 4 \* red LED
* 1 \* micro-USB cable

**How to make it**

**Preliminary**

1. Install MicroPython interpreter onto your Pico. Follow this guide by Raspberry Pi on how to do that <https://www.raspberrypi.com/documentation/microcontrollers/micropython.html#drag-and-drop-micropython>
2. Next, install Thonny so we can write our code. <https://thonny.org/>

A screenshot of a computer

Description automatically generated with low confidence

Figure 1 - Pico Schematic

**On your breadboard:**

1. Insert your Pico into your breadboard with the pins either side of the divide down the length of the breadboard. Insert the pins on the micro-USB end of the Pico into the 4th row on your breadboard (labelled 4); the bottom pins will be connected to row 23 if done correctly.
2. Now we’ll connect the button. Insert one of the ends of your jumper cable into A22 on your breadboard (this correlated to the GP14 pin on the Pico – see Figure 1 - Pico Schematic). Now connect the other end to D29. Connect your button across the centre divide such that its legs are in the holes E29, E31, F29 and F31. Connect another jumper cable to G31 and the other end to the red (positive) power-rail above the 31st row i.e. above the centre divide. We now connect one last jumper cable from the red power-rail above the centre divide back to J8 (check that this corresponds to the 3V3(OUT) pin Figure 1 - Pico Schematic).
3. Connecting the LEDs.
   1. LED 1: Connect a jumper cable from A18 (GP10) to C36. Then connect a 220 ohm resistor from D36 to D46. Connect the longer leg of the LED to E46 and the other end across to F46. Connect a jumper from F46 to the blue power-rail of row 46.
   2. LED 2: Connect a jumper from A19 to B37. Connect a resistor from C37 to C49. Connect the longer leg of the LED to E49 and the other across the divide to F49. Connect a jumper from G49 to the blue power-rail on the same row.
   3. LED 3: Connect a jumper from A20 to B35. Connect a resistor from C25 to C43. Connect the longer leg of the LED from E43 to F43 and a jumper from G43 to the blue power-rail on the same row.
   4. LED 4: Connect a jumper from A23 to A34. Connect a resistor from B34 to B40. Connect the longer leg of the LED from E40 to F40. Connect a jumper from G40 to the blue power-rail on the same row.
   5. Grounding: Connect a jumper from row 34 of the blue power-rail to I21 (GND pin).
4. Drink some water, you’re done! Note: don’t spill it on your new machine!!

**How to run it**

Open the code supplied with this document in Thonny and simply click ‘run’ and then type into the shell next to ‘>>>’ either: *auto\_sequence(0.3)* or *button\_sequence()*

*Have fun!*

*Harvey :)*