Measurement and Hardware timing

Week no. – 2

Date of experiment – 29/08/2023

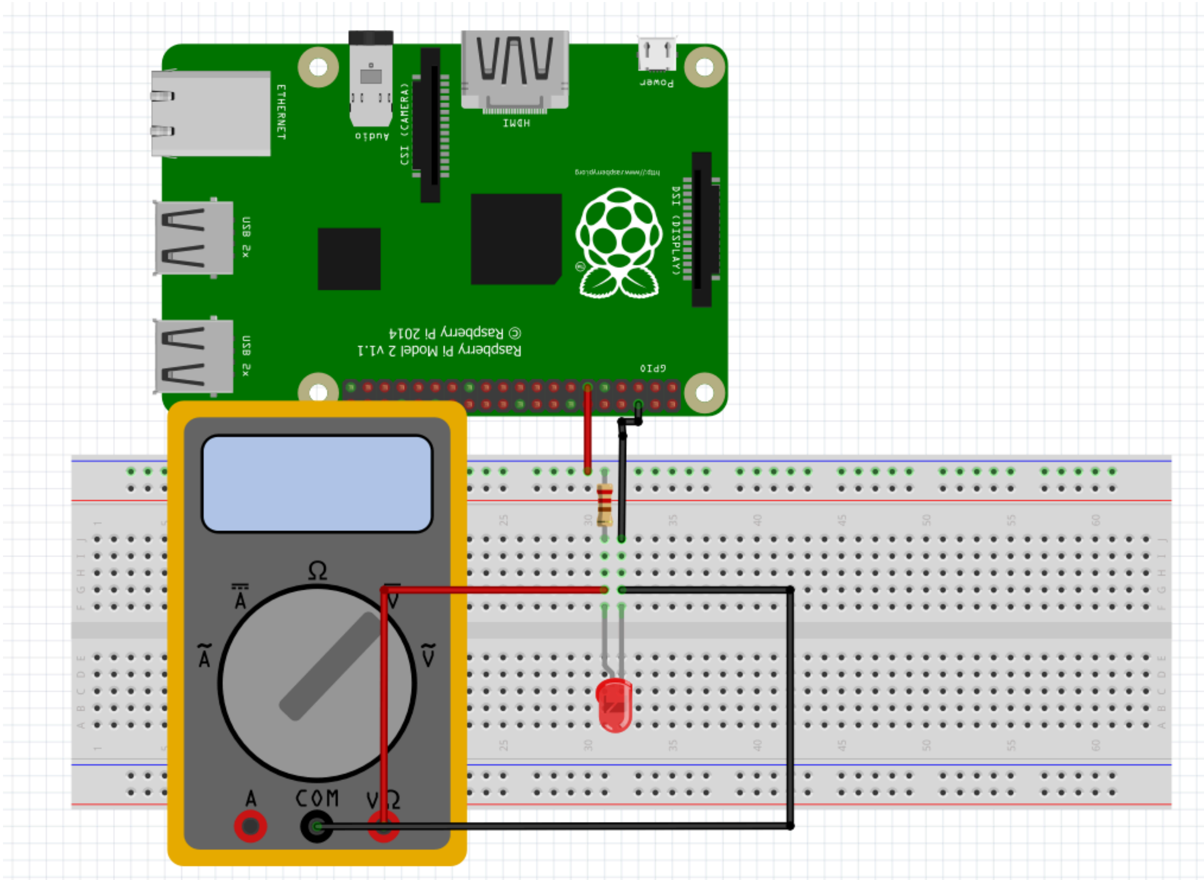
Group name – Group 6

Group members – Binduthraya Matta (B21203), Saransh Duharia (B21021), Jasnoor Tiwana (B21194), Rawal Ram(B21219), Adarsh Santoria (B21176).

---

1. Experiment layout

Following is the circuit diagram of the experiment.



SOFTWARE USED: FRITZING

Steps involved:

Setup: We set up the connections as that of lab 1. We connected the positive leg of an LED through a current limiting resistor (470 Ω) to one of the GPIO pins, and the negative leg to a ground GPIO pin, as in the last lab. Connect the Oscilloscope such that you can measure the voltage at the GPIO pin.

Writing the Program: We wrote a code similar to that of Lab 1. The objective was to blink the LED with a certain frequency for a certain period of time.

Using the Oscilloscope: We attached the positive and ground pins of the Oscilloscope to the terminals of the LED bulb to measure the output voltage. Later we adjusted the range and value/grid of X and Y, which were voltage across LED and time respectively, using knobs to get the desired output. The recorded waveform was adjusted in such a way so that only one rising or falling edge would be visible on the screen, in order to properly record the jitter. The cursors were adjusted to measure that jitter.

2. Results (in order of the indicated task completion criteria)

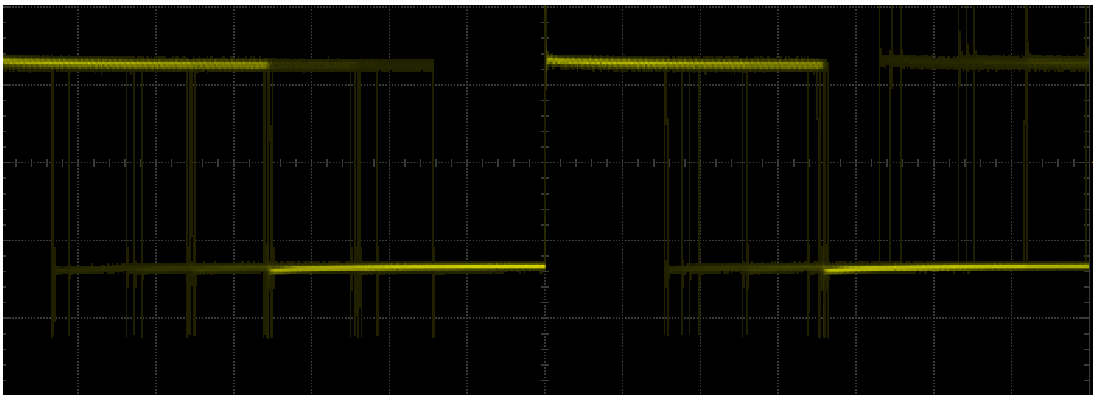
Task 1

Rewrite the code from Lab 1, such that you have two control variables: light_on which allows you to control the duration for which the LED is actively blinking and "frequency" which allows you to control the duration T of a single period. Run the LED for 2 s and repeat the experiment with different blinking frequencies. Observe the jitter on the oscilloscope (adjust the horizontal window to analyse 10 periods). Repeat the experiment by varying the frequency and categorize your observations for the jitter strength (very strong, strong, medium, weak, not observable). Repeat the experiment while you play any game on the RPi in parallel.

TABLE:

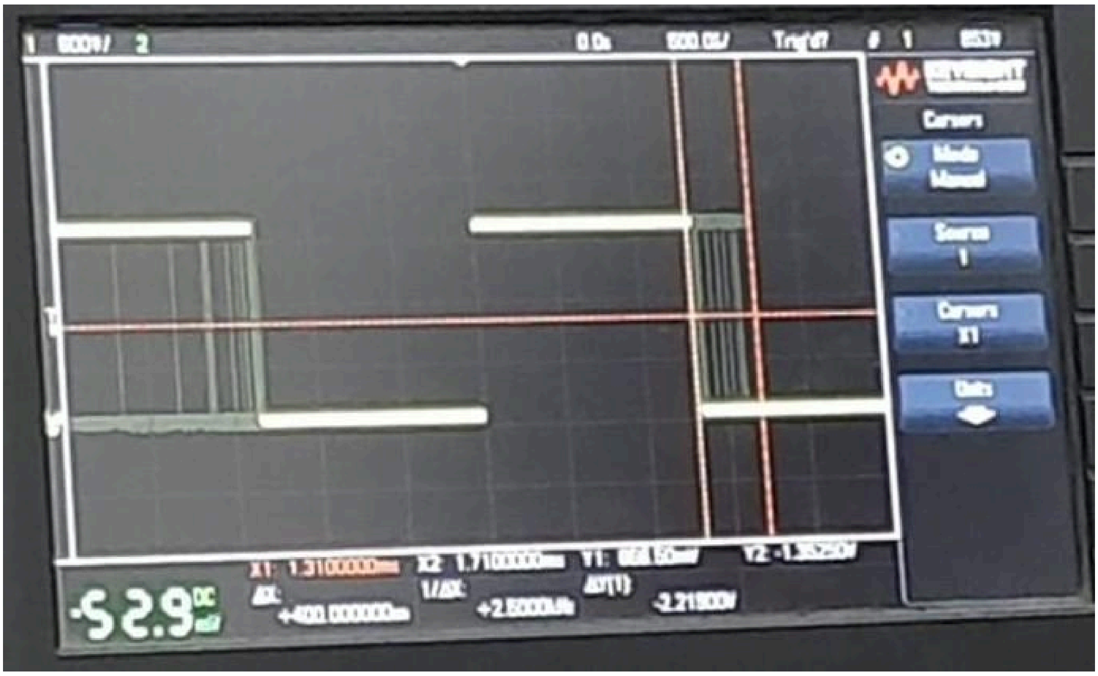
Frequency	Jitter strength w/o game	Jitter strength with parallel game	Average jitter measured by Oscilloscope
10	Not Observable	Weak	0
400	Slightly Visible/Weak	Slightly Visible/Weak	231us
1000	Medium	Medium	126us
5000	Strong	Strong	128us
10000	Starts overlapping/Strong	Starts overlapping/Strong	82us
100000	Completely Overlaps/Very Strong	Completely Overlaps/Very Strong	117us

We can see from the above table that without a parallel game running, the LED blinking remained stable and consistent at lower frequencies (10 Hz), with no observable jitter.



Some extra impression of plot visible (Jitter)

Jitter became noticeable as the LED blinking frequency increased. This suggests that timing precision may be affected as the frequency approaches and exceeds 1000 Hz. The results demonstrate the importance of managing system resources and considering timing constraints in real-time systems or applications requiring precise timing. We also notice here that when we are increasing frequency, jitter initially increases due to increase in jitter strength but it also decreases after some point because of decreasing time period of oscillation. These findings are valuable for fields such as embedded systems, robotics, and industrial automation, where stable timing is critical for system performance.



Slightly Visible/Weak Jitter of 231μs in 400Hz frequency

3. Conclusion

1. **What in your opinion is the most important thing you learnt from the experiment?**
   The most important thing learned from this experiment is how varying the blinking frequency of an LED affects the observed jitter on an oscilloscope and how it might be influenced when running other processes concurrently on the Raspberry Pi.

2. **What was interesting about the experiment?**
   The interesting aspect is observing how changing the LED blinking frequency affects the stability of the signal, and how external factors, such as running other processes on the Raspberry Pi, can introduce jitter in the signal.

3. **What was challenging about the experiment?**
   The challenging part may have been accurately measuring and categorizing the jitter strength and ensuring that external factors, like parallel processes, were consistent in their impact on the experiment. Sometimes, the jitter of two cycles would collapse making it hard to find the range of jitter.

4. **Were there any drawbacks of the way the experiment was done? How would you do it better?**
   One potential drawback could be the influence of other system processes on the Raspberry Pi, which may not provide consistent results. To mitigate this, you could consider isolating the Raspberry Pi or using a real-time operating system (RTOS) for more accurate measurements.

5. **How do you think it links with the next experiment?**
   The knowledge gained from this experiment about jitter and signal stability can be valuable in the context of subsequent experiments. For example, it could be relevant when working with sensors or communication systems where signal stability is crucial.

6. **Contribution statement –** In our experiment, each team member played a specific role to ensure its success:
   Adarsh Santoria: Conducted circuit analysis, helped build the circuit, and assisted in data collection.
   Binduthraya: Managed the Raspberry Pi setup and executed the code, ensuring the LED blinked according to the specified parameters.
   Saransh Duharia: Assisted in setting up the oscilloscope, capturing data, and categorizing observations regarding jitter strength.
   Jasnoor: Helped in Circuit execution, did coding of Raspberry Pi and Analyzed the collected data to visualize the results.
   Rawal Ram: Coding and Helped in Oscilloscope execution.
   Collectively, our team worked together to design and execute the experiment, ensuring a comprehensive analysis of the results and a well-rounded understanding of the experiment's outcomes.