Toolchain: Full Circle Latency

Software Engineering for IoT

Markus Solomon Benjamin Barrow – mabar16
Victor Steenfeldt Laursen – vilau16

1 Experimental Setup

This experiment makes use of two PyCom devices. The first device is programmed to blink for one second at a frequency of one blink per second, the light emitter. The second device is set to detect changes in light levels every 50 ms with the light sensor. In order to ensure stability in the physical setup, the devices are attached to each other using a rubber band, such that the light emitting device's LED is directly facing the light sensing device's light sensor.

Both devices are connected to the same laptop via USB. The laptop runs two Python programs, one for each connected device. The first Python program sends alternating ones and zeroes every second to the light emitter. Upon receiving a 1 it activates its LED and upon 0 it deactivates the LED. The laptop logs the time stamp before and after sending a value to the light emitter.

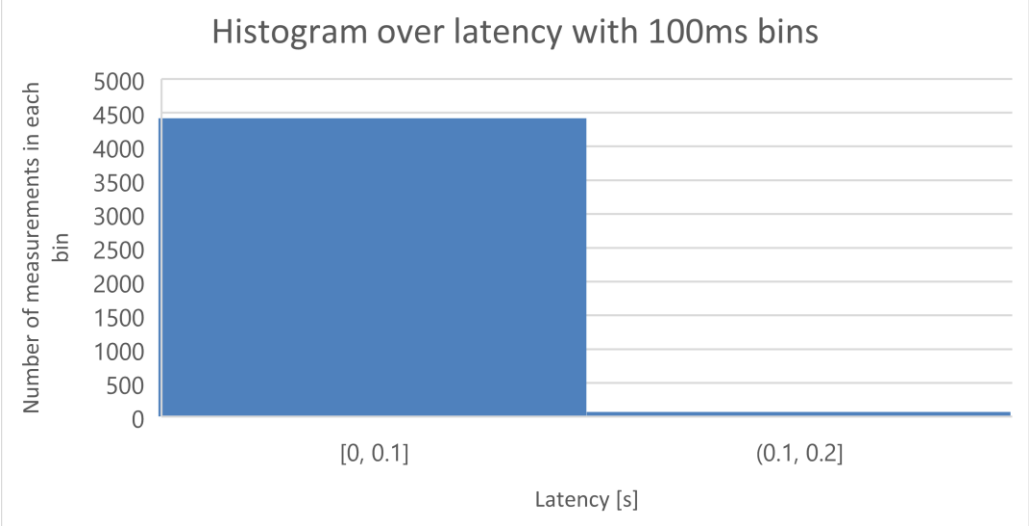
The light sensor sends a 1 to the laptop whenever it detects the light emitter's LED activating and sends a 0 when it deactivates. The laptop logs the time stamp immediately before and after each digit is received.

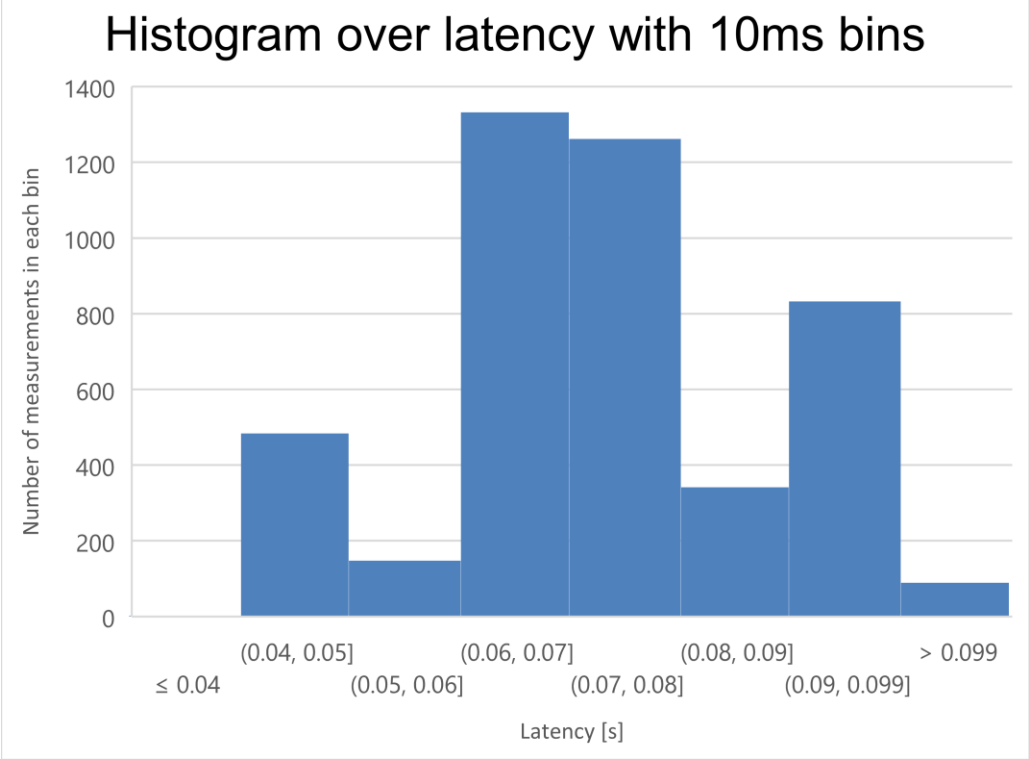
This allows the latency from light emission to detection to be calculated as the difference between the time stamp logged before the laptop sends a 1 to the light emitter and the time stamp logged after the laptop receives a 1 from the light sensing device.

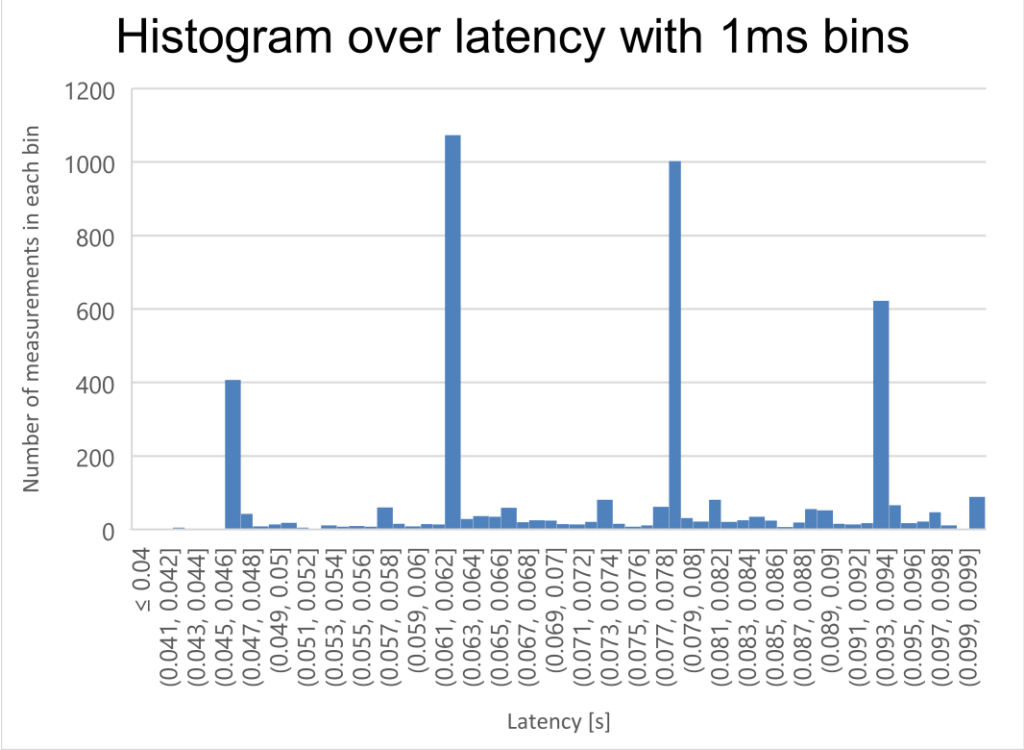
The light sensing device uses the following configuration settings: 96x gain, 50ms integration time, and 50ms between measurements. It uses a threshold of 10000 lux to determine whether the LED is on. This threshold has been chosen as it was observed that the sensor measured 12000~13000 lux when the LED was on.

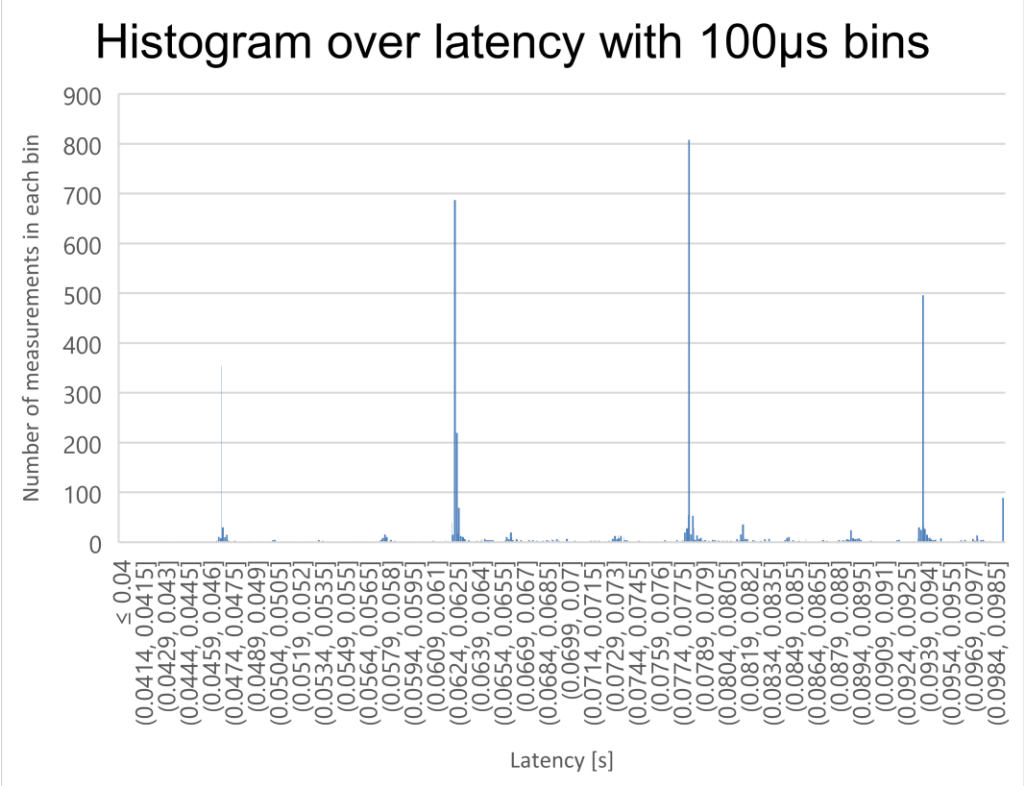
2 Results

This section shows four histograms of the measured latencies. They are discussed in Section 3.






Histogram over latency with 1ms bins
Number of measurements in each bin
1200
1000
800
600
400
200
0
≤ 0.04
(0.041, 0.042]
(0.043, 0.044]
(0.045, 0.046]
(0.047, 0.048]
(0.049, 0.05]
(0.051, 0.052]
(0.053, 0.054]
(0.055, 0.056]
(0.057, 0.058]
(0.059, 0.06]
(0.061, 0.062]
(0.063, 0.064]
(0.065, 0.066]
(0.067, 0.068]
(0.069, 0.07]
(0.071, 0.072]
(0.073, 0.074]
(0.075, 0.076]
(0.077, 0.078]
(0.079, 0.08]
(0.081, 0.082]
(0.083, 0.084]
(0.085, 0.086]
(0.087, 0.088]
(0.089, 0.09]
(0.091, 0.092]
(0.093, 0.094]
(0.095, 0.096]
(0.097, 0.098]
(0.099, 0.099]
Latency [s]


Histogram over latency with 100µs bins
Number of measurements in each bin
900
800
700
600
500
400
300
200
100
0
≤ 0.04
(0.0414, 0.0415]
(0.0429, 0.043]
(0.0444, 0.0445]
(0.0459, 0.046]
(0.0474, 0.0475]
(0.0489, 0.049]
(0.0504, 0.0505]
(0.0519, 0.052]
(0.0534, 0.0535]
(0.0549, 0.055]
(0.0564, 0.0565]
(0.0579, 0.058]
(0.0594, 0.0595]
(0.0609, 0.061]
(0.0624, 0.0625]
(0.0639, 0.064]
(0.0654, 0.0655]
(0.0669, 0.067]
(0.0684, 0.0685]
(0.0699, 0.07]
(0.0714, 0.0715]
(0.0729, 0.073]
(0.0744, 0.0745]
(0.0759, 0.076]
(0.0774, 0.0775]
(0.0789, 0.079]
(0.0804, 0.0805]
(0.0819, 0.082]
(0.0834, 0.0835]
(0.0849, 0.085]
(0.0864, 0.0865]
(0.0879, 0.088]
(0.0894, 0.0895]
(0.0909, 0.091]
(0.0924, 0.0925]
(0.0939, 0.094]
(0.0954, 0.0955]
(0.0969, 0.097]
(0.0984, 0.0985]
Latency [s]

3 Analysis

In this section, we compare the actual and expected distributions of latency.

3.1 Expected distribution of latency

We expect a normal distribution of latency with a mean around 50ms and a low standard deviation.

We reason that the latencies should be normal distributed, because we expect that they will all be close to the mean and any differences in latency caused by the time taken by communication, the time needed for the LED to turn on, or time taken by the light sensor will be on both sides of the mean.

50ms has been chosen as the mean because that is the interval between each sensing of the light sensor. That is, it is able to perform measurements with a frequency of 1/50ms.

3.2 Measured distribution of latency

The actual distribution of latencies is shown in the histograms shown in Section 2. The histogram with bins of 100ms is not very useful. It only shows that the vast majority of measurements are below 100ms.

The histogram with 10ms bins shows that most of the latencies measured are around 70ms. Looking at this histogram one is tempted to conclude that the latencies are normally distributed.

The next histogram (with 1ms bins), however, shows that the latencies do not follow a normal distribution. In fact, they do not follow any of the common statistical distributions (normal, uniform, exponential, etc.). Most of the latencies are distributed amongst only four bins.

The histogram with 100µs bins supports this. It shows four tall bars representing ~73% of all the measurements. They are about 16ms apart.

3.3 Deviations

We are not certain why the latencies are distributed like this, but we have a suspicion that it is due to the USB/Serial interface on the Windows laptop. An answer on Stack Overflow (stackoverflow.com/a/55244435/7063253) mentions the following: "*On Linux & Windows, the default latency timer setting is 16ms. For example, say you send a 3 byte MIDI message from your Arduino at 115200bps. As serial data, it takes 0.3ms for the MIDI message to go from the Arduino's microcontroller to the FTDI chip. However, the FTDI holds the message in its buffer for a further 15.8ms (16ms after the first byte arrived), before the latency timer expires and it sends a USB packet to the computer.*"