1 Measurements and Tests

The measurements and tests cover the power drain from the sensor, laser and the esp32l inlcuding deepsleep and bluetooth, to determine the optimal setup. The setup for the measurement or test is explained in each section. Again, the tests do not aim for scientific and statisitcal significance but as a guiding point. Their goal is to provide an order of magnitude of the power consumption of the individual components to guide future architectural decisions.

1.1 Laser

We measured the laser on a power source directly connected to the esp32 which has an output rating of $4.695V$ with a max current of $80mA$. The laser consumed on average $25mA$ and thus the power consumption is

$$4.695V * 25mA = 0.117W$$

which is, as expected, fairly high. In a setup with to lasers per goal, this could be a realy problem

1.2 Sensor

Next, we measured the power consumption of the sensor, again powered by the esp32, so $4.695V$ with a max current of $80mA$. It consumed on average $3.5mA$ when the laser was hitting the sensor and $3.8mA$ when it was not. As the default state is hitting and only goals, so very short periods, are not hitting, the power consumption is

$$4.695V * 3.5mA = 0.016W.$$

This is inline with the expectations, as the resistance drops when the laser hits the sensor. To summarize, together the sensor and laser consume around

$$0.016W + 0.117W = 0.133W$$

in idle and a little bit more when a goal is scored. Thus the power consumption in one day is

$$0.133W * 24 = 3.19Whr$$

This is quite a lot especially running on a battery with only

$$0.133W * 24 = 12Whr$$

1.3 ESP32 Modes

The esp32 development board from wemos consumes around $43mA$ in idle and $0.128mA$ in deepsleep. This is much higher than esp32 wroom board which is supposed to only consume $20mA$ in idle and $0.01mA$ in deepsleep[1]. Figure 1 shows both microcontrollers side by side.

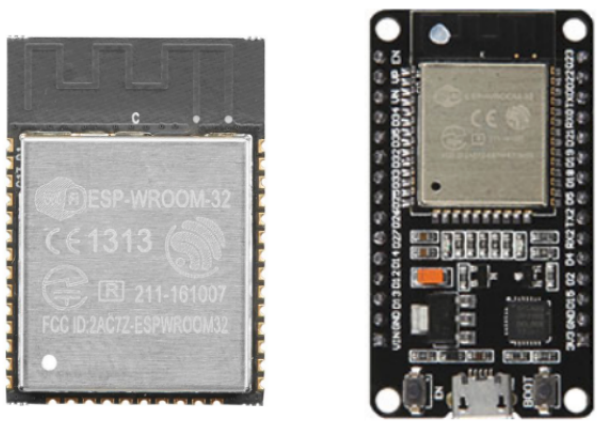

Figure 1: ESP32 wroom (left) and Wemos development board.

The development board has a microusb controller and additional current controls and a power LED. These extra components add up to the significant power consumption difference. For the production we thus plan on using a esp32 Wroom and soldier the connections ourselfs. So instead of using our numbers, we will use the numbers from the esp32 wroom controller[1].

The power consumption also changes during boot time and during data transmission. In the former, the processor consumes a lot of power, while in the latter, the radio chip, including antenna, consumes the majority of extra power. Unfortuately, we were not able to get a esp32 wroom in time for the tests, so we have to infere values from the esp32 wemos development board.

In our tests waking up from deepsleep took around $300ms$ and increased current consumption to about $80mA$[1]. Especially, the wake up time seems quite slow and we would like to test this further and search for wake up optimizations in further research. During sending and receiving of data, the power consumption increased to $70mA$.

To summarize each wake up costs at least

$$0.08A * 5V/3600 = 0.00037Wh = 0.37mWh$$

so for a startup time of 1/3 and sending data costs

$$0.07A * 5V/3600 = 0.00037Wh = 0.32mWh$$

[1] This was done via eye measurment with a lot of fluctations, so the numbers are really only approximates.

1.4 BLE Tests

References

[1] *Insight into esp32 sleep modes & their power consumption*, `https://lastminuteengineers.com/esp32-sleep-modes-power-consumption/`, (Accessed on 05/28/2019).